Lesson Module Status

- Slides – draft
- Properties - done
- Flashcards - na
- 1st minute quiz – done
- Web Calendar summary – done
- Web book pages – done
- Commands –
- Howtos –
- Skills pacing -na
- Lab – done
- Depot (VMs) – restored

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/


Joe A.

Joe P.

Teach & Confer is a live interactive classroom to meet with your students.
STUDENT LOG IN
View Teach & Confer Archives

www.cccconfer.org
dial-in: 888-886-3951
passcode: 439080


John


Chris B.


Chuck

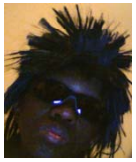

Rich


Josh

Robert


Casady

Chris H.


Lieven

Jesus


Jack

Edwin

Julio


Drew


Edgar

Kay


Ryan

Aaron


Joe B.

Junious

Brynden

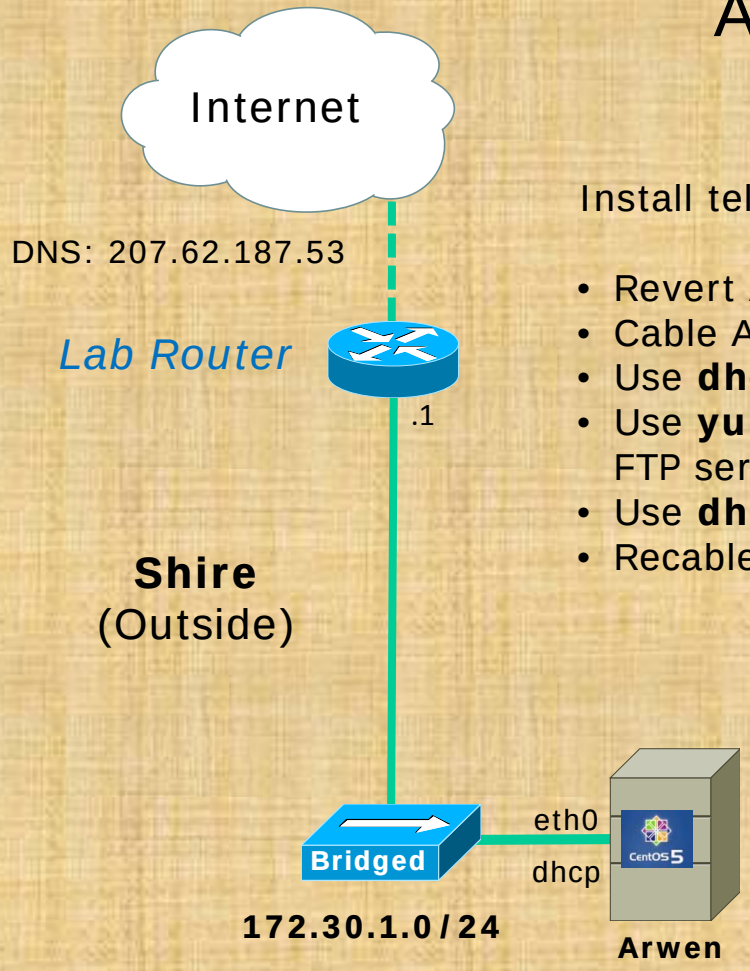
Email me (risimms@cabrillo.edu) a relatively current photo of your face for 3 points extra credit

Quiz

Please take out a blank piece of paper, switch off your monitor, close your books, put away your notes and answer these questions:

- How do you find out if vsftpd is installed?
- What two ports does FTP use?
- What command shows the ports on your system that are open and listening for requests?

Activity



Install telnet-server and vsftpd on Arwen:

- Revert Arwen to its snapshot
- Cable Arwen to the classroom network (bridged)
- Use **dhclient eth0** to get an IP address
- Use **yum install telnet-server vsftpd** to install Telnet and FTP server applications
- Use **dhclient –r eth0** to release the IP address
- Recable to VMnet3

After installing packages:

```
[root@arwen ~]# rpm -qa | grep telnet
telnet-0.17-39.el5
telnet-server-0.17-39.el5
[root@arwen ~]# rpm -qa | grep vsftpd
vsftpd-2.0.5-16.el5_4.1
[root@arwen ~]#
```

Firewalls and NAT

Objectives	Agenda
• Configure a network service with security restrictions for its use using either TCP Wrappers or a superdaemon. • Use iptables to build a permissive firewall by selectively filtering packets based on protocol type. • Create a secure tunnel between two hosts that allows port forwarding into a private network. • Use Network Address Translation (NAT) to allow hosts on a private network to access the Internet.	• Quiz • Questions on previous material • Scripting network setting changes • Housekeeping • Wrap up transport layer • Application Layer • Super daemons • TCP wrappers • Telnet • FTP • SSH • SSH port forwarding • Example firewall and NAT • Netfilter • Lab 5 Prep • Wrap

Questions on previous material

Questions?

- Previous lesson material
- Lab assignment

Housekeeping

- Lab 4 due today!
- MSDN AA success?
- Test 1 graded, still a few students still need to take it before we can go over results
- Mike Brogan from Cruzio on May 25th
- Use the forum for any MSDN AA issues
- Hands-on cabling using system pods

- Revert and power up Frodo, Elrond and Arwen VMs for tonight

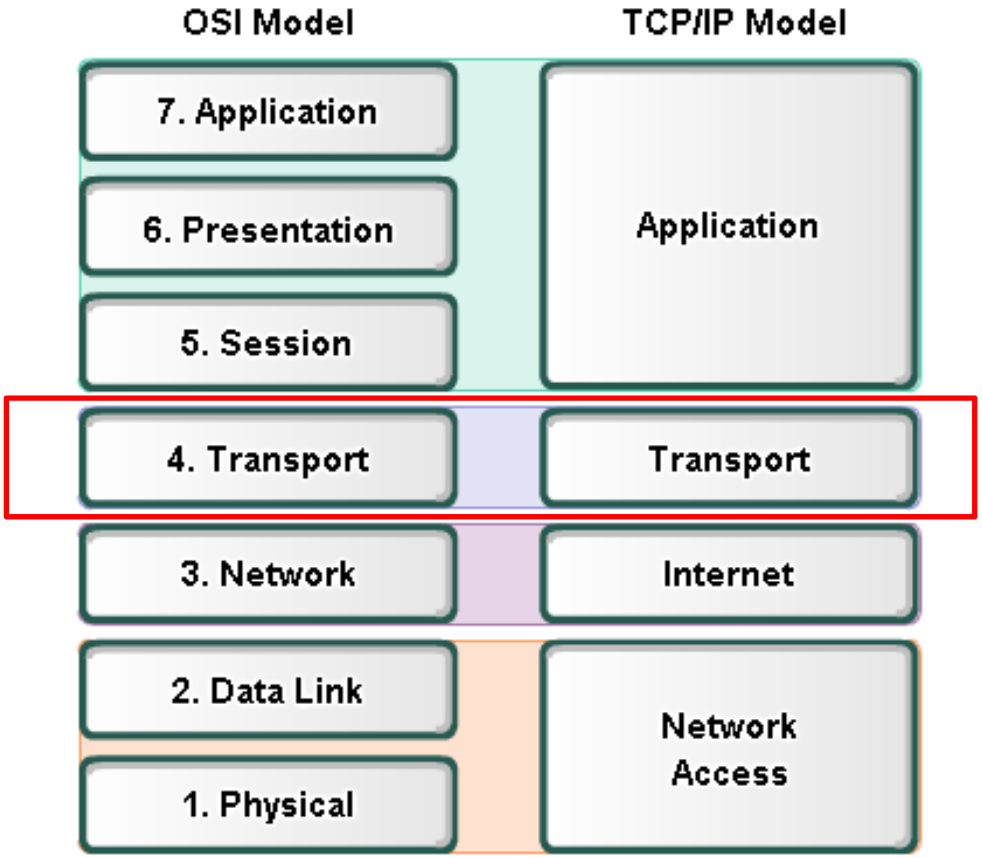
Finish up TCP

Transport Layer

Objectives:

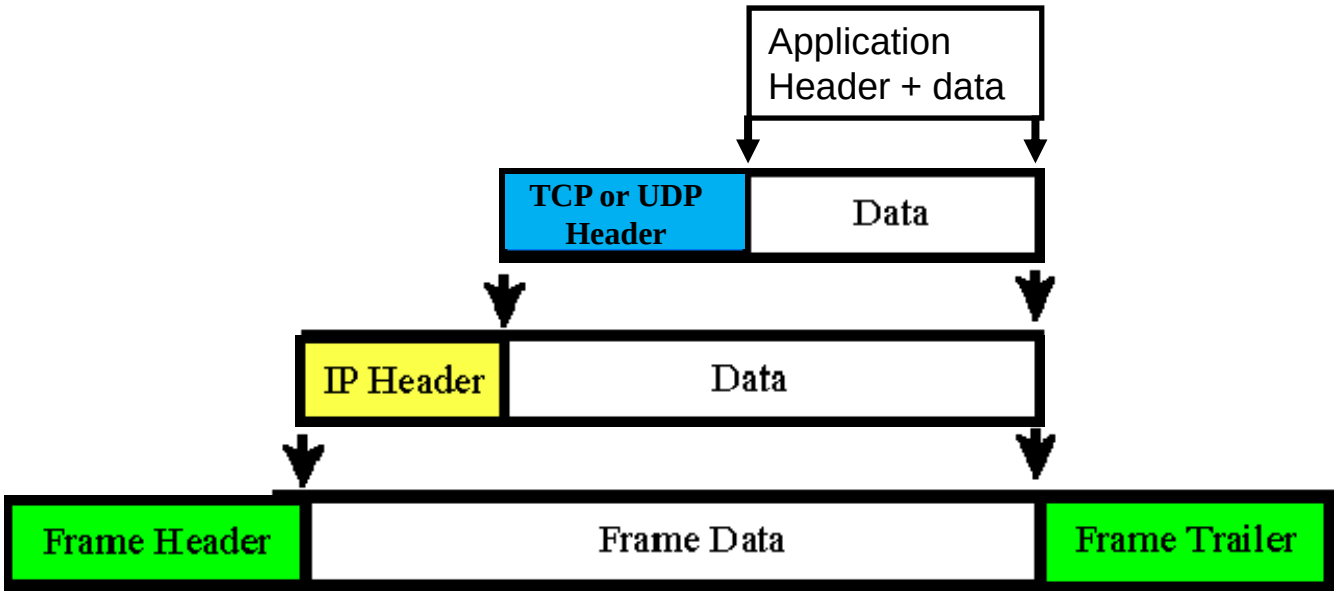
- Understand key TCP header information
- Understand how TCP connections are made and ended
- Understand how a socket is defined
- Recognize connection state changes and sockets by looking at Wireshark captures

Protocol and Reference Models



- The **Open Systems Interconnection (OSI)** model is the *most widely known internetwork reference model*.

Transport Layer



Application layer

Layer 4 (Transport)

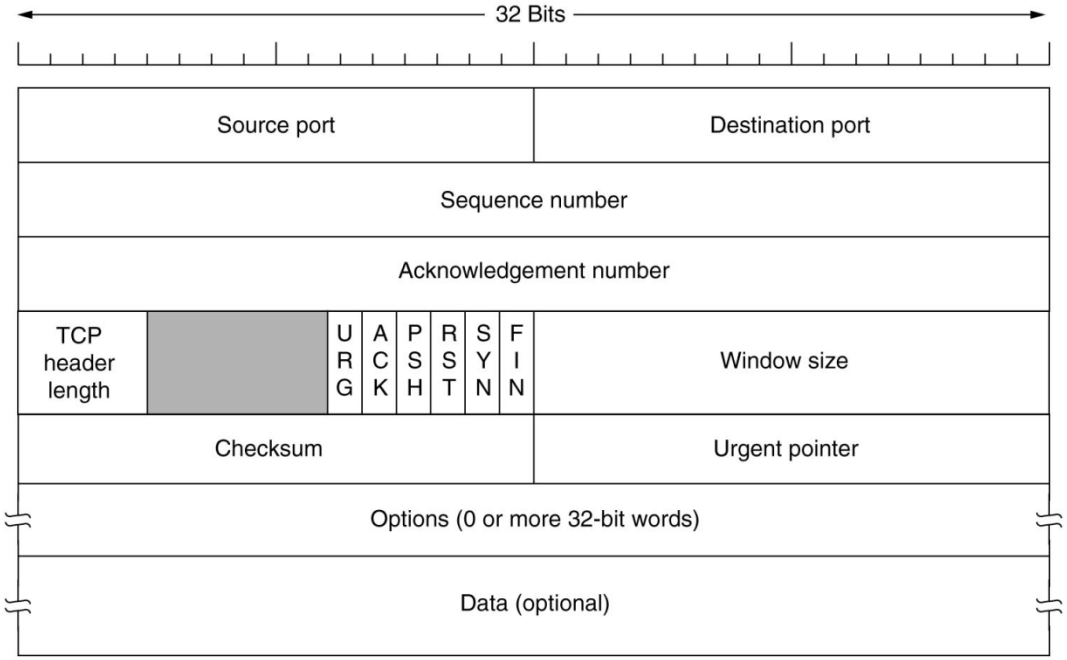
Layer 3 (Network)

Layer 2 (Data Link)

Transport Layer

The Transmission Control Protocol

TCP Header



Ports *are used to identify application*

Sequence *and* ***acknowledgement numbers*** *are used for flow control.*

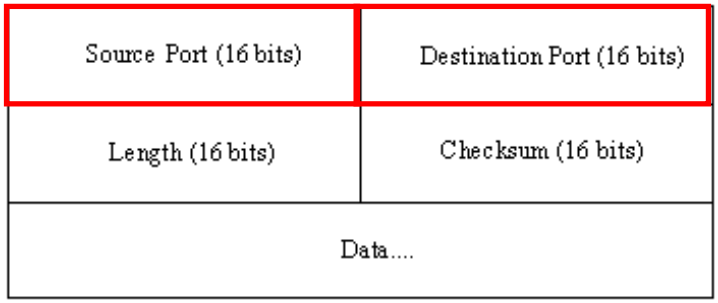
ACK, ***SYN*** *and* ***FIN*** *flags are used for initiating connections, acknowledging data received and terminating connections*

Window size *is use to communicate buffer size of recipient.*

Options *like SACK permit selective acknowledgement*

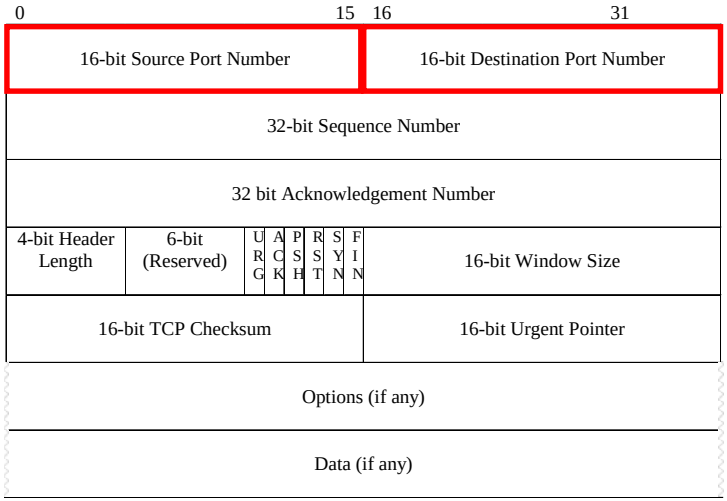
Data *contains application specific information*

UDP Header



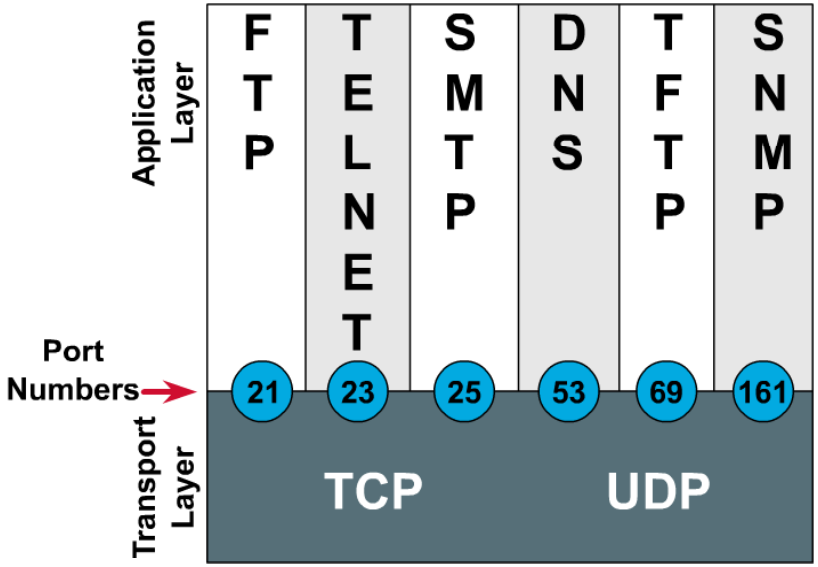
User Datagram Protocol (UDP)
= Connectionless , Stateless , Unreliable

TCP Header



Transmission Control Protocol (TCP)
= Connection-oriented ,Stateful, Reliable

Port Numbers



E.g. HTTP is Port 80

Transport Layer

Sockets

Sockets are communication endpoints which define a network connection between two computers (RFC 793).

- Source IP address
- Source port number
- Destination IP address
- Destination port number

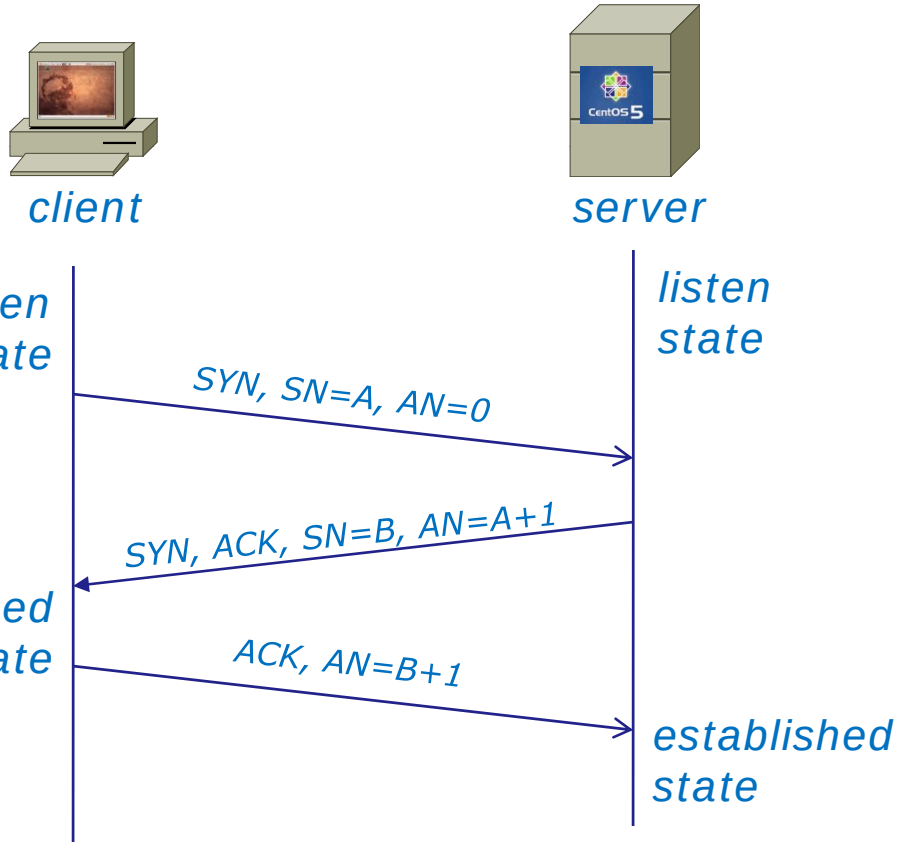
A socket is uniquely defined by the source IP address, source port, destination IP address, and destination port

Transport Layer

Initiating a new TCP Connection

Three-Way Handshake

1. SYN
2. SYN-ACK
3. ACK



AN=Acknowledgment Number
SN=Sequence Number
ACK=ACK flag set
SYN=SYN flag set

Transport Layer

The Transmission Control Protocol

Continuing communications on an established connection

- The Sliding Window
  Used for flow control - allows sending additional segments before an acknowledgement is received based on recipients buffer size

- Flow Control (cumulative acknowledgment)
  Recipient tells sender the size of its input buffer and sends acknowledgements when data has been received. Sequence numbers are used to detect missing segments.

- The SACK option
  Selective acknowledgement so only the dropped segments need to be retransmitted.

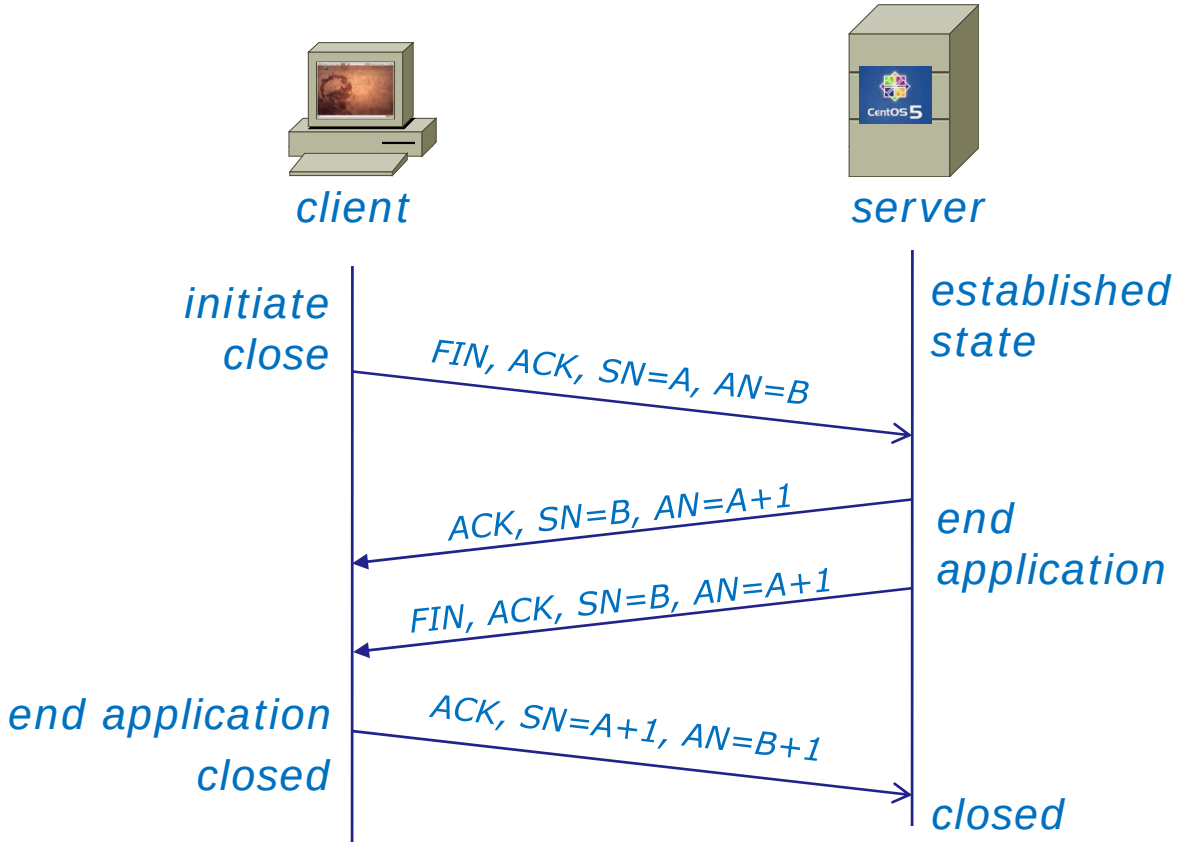
- The RST Flag
  Used to terminate a connection when an abnormal situation happens

Transport Layer

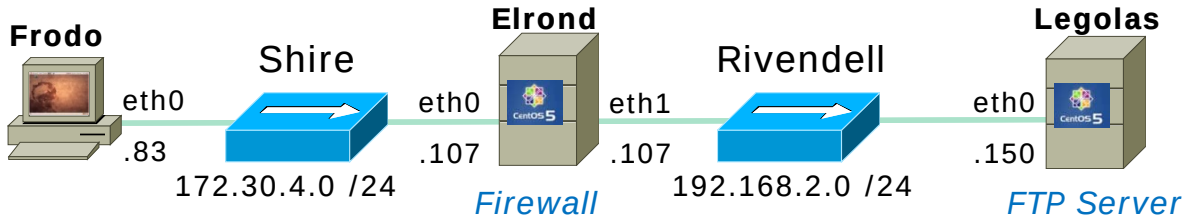
Closing a TCP Connection

Four-Way Handshake

1. FIN, ACK
2. ACK
3. FIN, ACK
4. ACK



AN=Acknowledgment Number
SN=Sequence Number
ACK=ACK flag set
FIN=FIN flag set



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

Socket for data transfer

Client	Server
172.30.4.83	192.168.2.150
42571	20

```
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=1
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

Retrieve legolas file

3 way handshake initiated by server

File transfer

4 way handshake to close connection

Tunable Kernel Parameters

Exercise

Explore the TCP, UDP and other variables in the **/proc/sys/net/ipv4** directory

```
[root@bigserver ~]# ls /proc/sys/net/ipv4
cipso_cache_bucket_size           ip_dynaddr              tcp_dsack             tcp_retries2
cipso_cache_enable                ip_forward              tcp_ecn               tcp_rfc1337
cipso_rbm_optfmt                  ipfrag_high_thresh      tcp_fack              tcp_rmem
cipso_rbm_strictvalid             ipfrag_low_thresh       tcp_fin_timeout       tcp_sack
conf                              ipfrag_max_dist         tcp_frto              tcp_slow_start_after_idle
icmp_echo_ignore_all              ipfrag_secret_interval  tcp_keepalive_intvl   tcp_stdurg
icmp_echo_ignore_broadcasts       ipfrag_time             tcp_keepalive_probes  tcp_synack_retries
icmp_errors_use_inbound_ifaddr    ip_local_port_range     tcp_keepalive_time    tcp_syncookies
icmp_ignore_bogus_error_responses ip_nonlocal_bind        tcp_low_latency       tcp_syn_retries
icmp_ratelimit                    ip_no_pmtu_disc         tcp_max_orphans       tcp_timestamps
icmp_ratemask                     neigh                   tcp_max_syn_backlog   tcp_tso_win_divisor
igmp max memberships              netfilter               tcp max tw buckets    tcp tw recycle
igmp_max_msf                      route                   tcp_mem               tcp_tw_reuse
inet_peer_gc_maxtime              tcp_abc                 tcp_moderate_rcvbuf   tcp_window_scaling
inet_peer_gc_mintime              tcp_abort_on_overflow   tcp_mtu_probing       tcp_wmem
inet_peer_maxttl                  tcp_adv_win_scale       tcp_no_metrics_save
tcp_workaround_signed_windows
inet_peer_minttl                  tcp_app_win             tcp_orphan_retries    udp_mem
inet_peer_threshold               tcp_base_mss            tcp_reordering        udp_rmem_min
ip_conntrack_max                  tcp_congestion_control  tcp_retrans_collapse  udp_wmem_min
ip_default_ttl                    tcp_dma_copybreak       tcp_retries1

[root@bigserver ~]# cat /proc/sys/net/ipv4/tcp_sack
1
[root@bigserver ~]# cat /proc/sys/net/ipv4/tcp_syn_retries
5
[root@bigserver ~]#
```

Transport Layer

TCP Tunable Kernel Parameters

tcp_fin_timeout	*how long to keep in FIN-WAIT-2 state*
tcp_keepalive_time	*how long to keep an unused connection alive*
tcp_sack	*enable/disable selective acknowledgments*
tcp_timestamps	*enable RFC 1323 definition for round-trip measurement*
tcp_window_scaling	*enable RFC 1323 window scaling*
tcp_retries1	*how many times to retry before reporting an error*
tcp_retries2	*how many times to retry before killing connection*
tcp_syn_retries	*how many times to retransmit the SYN, ACK reply*

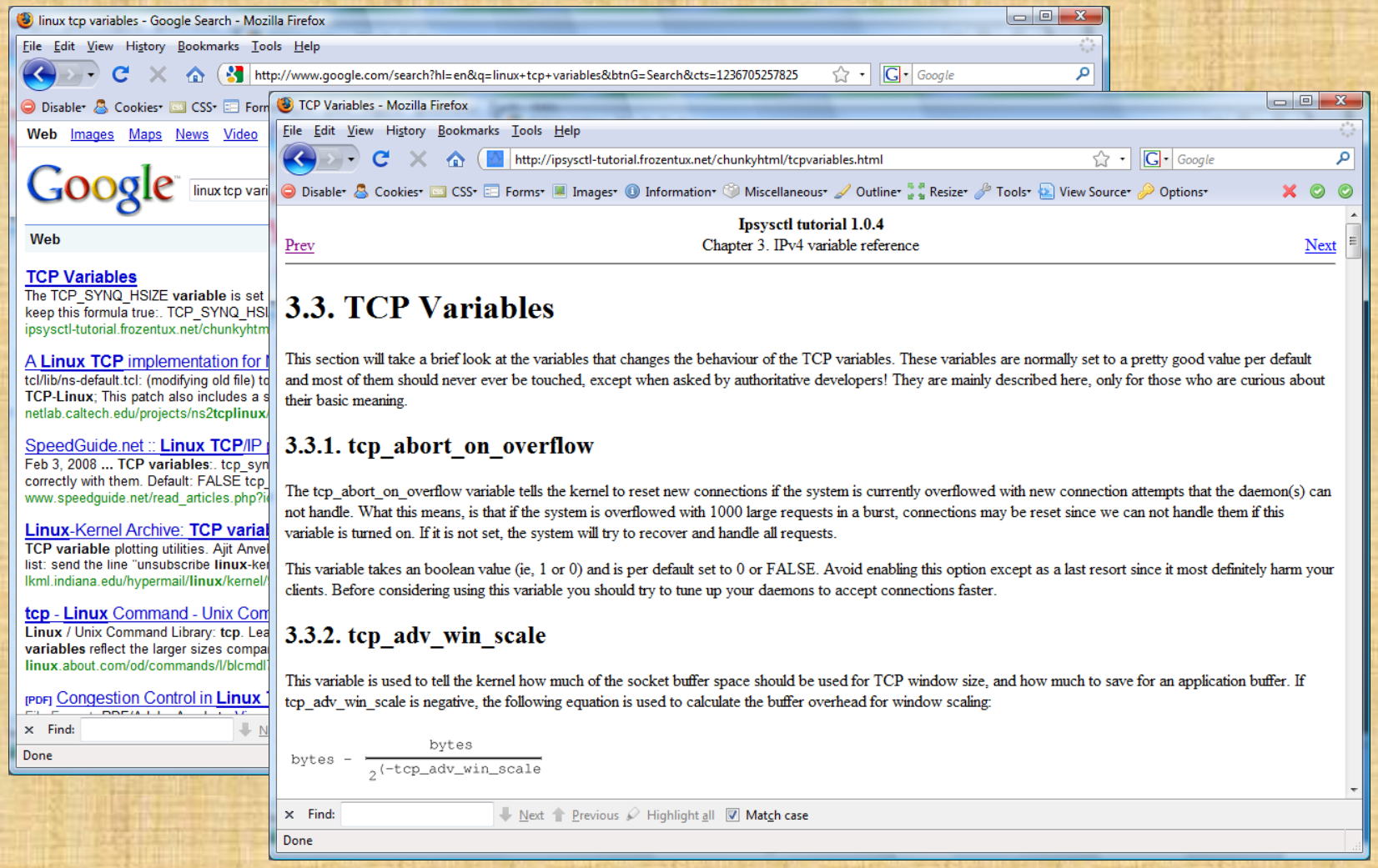
In the same directory:

ip_forward	*enable/disable selective acknowledgments*

```
[root@bigserver ~]# cat /proc/sys/net/ipv4/tcp_sack
1
[root@bigserver ~]# cat /proc/sys/net/ipv4/tcp_syn_retries
5
```

Exercise

Google linux tcp variables



Security Issues

Transport Layer

Inherent security vulnerabilities with TCP/IP protocols

- Denial of service attack using SYN flooding
- Falsifying TCP communications by spoofing, fake rip updates, bogus ping errors, malicious DNS entries, etc.
- Hijacking connections with sequence guessing, man in the middle attacks, etc.

Defenses: firewalls, authentication, and encryption

Tonight we will be looking at making firewalls

More:

- www.securityfocus.com
- www.linuxsecurity.com/resource_files/documentation/tcpip-security.html
- Jim's CIS 193 class

Warmup

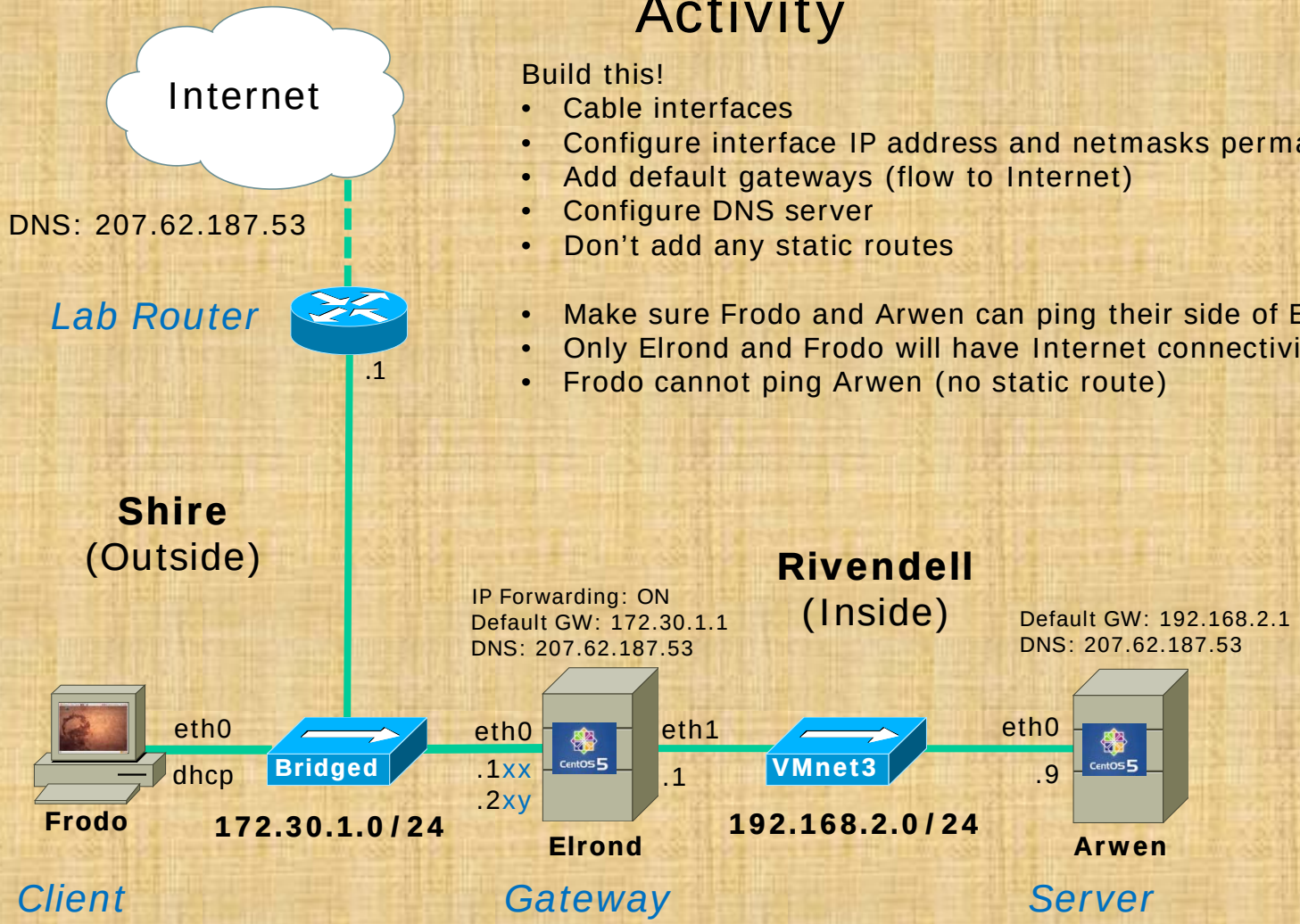
IP addresses for VM's in the classroom

Station	IP	Static 1
Instructor	172.30.1.100	172.30.1.125
Station-01	172.30.1.101	172.30.1.126
Station-02	172.30.1.102	172.30.1.127
Station-03	172.30.1.103	172.30.1.128
Station-04	172.30.1.104	172.30.1.129
Station-05	172.30.1.105	172.30.1.130
Station-06	172.30.1.106	172.30.1.131
Station-07	172.30.1.107	172.30.1.132
Station-08	172.30.1.108	172.30.1.133
Station-09	172.30.1.109	172.30.1.134
Station-10	172.30.1.110	172.30.1.135
Station-11	172.30.1.111	172.30.1.136
Station-12	172.30.1.112	172.30.1.137

Station	IP	Static 1
Station-13	172.30.1.113	172.30.1.138
Station-14	172.30.1.114	172.30.1.139
Station-15	172.30.1.115	172.30.1.140
Station-16	172.30.1.116	172.30.1.141
Station-17	172.30.1.117	172.30.1.142
Station-18	172.30.1.118	172.30.1.143
Station-19	172.30.1.119	172.30.1.144
Station-20	172.30.1.120	172.30.1.145
Station-21	172.30.1.121	172.30.1.146
Station-22	172.30.1.122	172.30.1.147
Station-23	172.30.1.123	172.30.1.148
Station-24	172.30.1.124	172.30.1.149

Note the static IP address for your station to use in the next class exercise

Activity



Build this!

- Cable interfaces
- Configure interface IP address and netmasks permanently
- Add default gateways (flow to Internet)
- Configure DNS server
- Don't add any static routes

- Make sure Frodo and Arwen can ping their side of Elrond
- Only Elrond and Frodo will have Internet connectivity
- Frodo cannot ping Arwen (no static route)

.1xx from http://simms-teach.com/docs/static-ip-addrs.pdf
.2xy (alias) is your station number + 200

Cheat Sheet

Elrond

/etc/sysconfig/network
```
NETWORKING=yes
NETWORKING_IPV6=no
HOSTNAME=elrond.localdomain
GATEWAY=172.30.1.1
```

/etc/sysconfig/network-scripts/ifcfg-eth0
```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
NETMASK=255.255.255.0
IPADDR=172.30.1.1xx
```

/etc/sysconfig/network-scripts/ifcfg-eth0:1
```
DEVICE=eth0:1
ONBOOT=yes
BOOTPROTO=static
NETMASK=255.255.255.0
IPADDR=172.30.1.2xy
```

/etc/sysconfig/network-scripts/ifcfg-eth1
```
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.2.1
NETMASK=255.255.255.0
```

/etc/resolv.conf
```
nameserver 207.62.187.53
```

/etc/sysctl.conf
```
< snipped >
net.ipv4.ip_forward = 1
< snipped >
```

Arwen

/etc/sysconfig/network
```
NETWORKING=yes
NETWORKING_IPV6=no
HOSTNAME=arwen.localdomain
GATEWAY=192.168.2.1
```

/etc/sysconfig/network-scripts/ifcfg-eth0
```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.2.9
NETMASK=255.255.255.0
```

/etc/resolv.conf
```
nameserver 207.62.187.53
```

Frodo

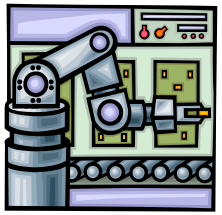
/etc/network/interfaces
```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
```

Restarting network service

Red Hat: **service network restart**

Ubuntu/Debian: **/etc/init.d/networking restart**

FYI section



Scripting using the **sed** (stream editor) command

This script modifies the permanent network settings files for the Elrond VM to work on the classroom network

```
#!/bin/bash
#
# Lab 5 - Classroom (Instructor station) Elrond VM
#
# Use classroom static IP address
#
sed -i '/IPADDR/ d' /etc/sysconfig/network-scripts/ifcfg-eth0
sed -i '$aIPADDR=172.30.1.125' /etc/sysconfig/network-scripts/ifcfg-eth0
sed -i '/IPADDR/ d' /etc/sysconfig/network-scripts/ifcfg-eth0:1
sed -i '$aIPADDR=172.30.1.200' /etc/sysconfig/network-scripts/ifcfg-eth0:1
#
# Use classroom gateway
#
sed -i '/GATEWAY/ d' /etc/sysconfig/network
sed -i '$aGATEWAY=172.30.1.1' /etc/sysconfig/network
#
echo Setting classroom network settings
service network restart
echo Interfaces:
ifconfig | grep -C1 eth
echo Gateway:
route -n | grep UG
```

identifies this file as a bash script

deletes any line with IPADDR

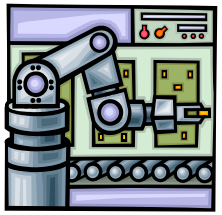
add new line to file

alias address file

using grep to show 1 line of context before and after matched line to reduce ifconfig output

using grep to just show gateways

Using sed: http://www.oracle.com/technology/pub/articles/dulaney_sed.html

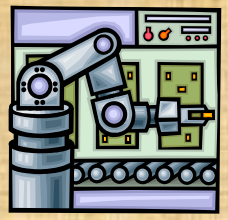


Scripting using the **sed** (stream editor) command

```
#
# Update NAT to new IP alias
#
iptables -t nat -R PREROUTING 1 -d 172.30.1.200 -i eth0 -j DNAT --to-destination 192.168.2.9
iptables -t nat -R POSTROUTING 1 -s 192.168.2.9 -o eth0 -j SNAT --to-source 172.30.1.200
iptables-save > /etc/sysconfig/iptables
service iptables restart
cat /etc/sysconfig/iptables | grep "DNAT\|SNAT"
```

using grep to show only lines containing DNAT or SNAT to reduce amount of output

escape the pipe character so it can be used as “or”



Example script to permanently change default gateway

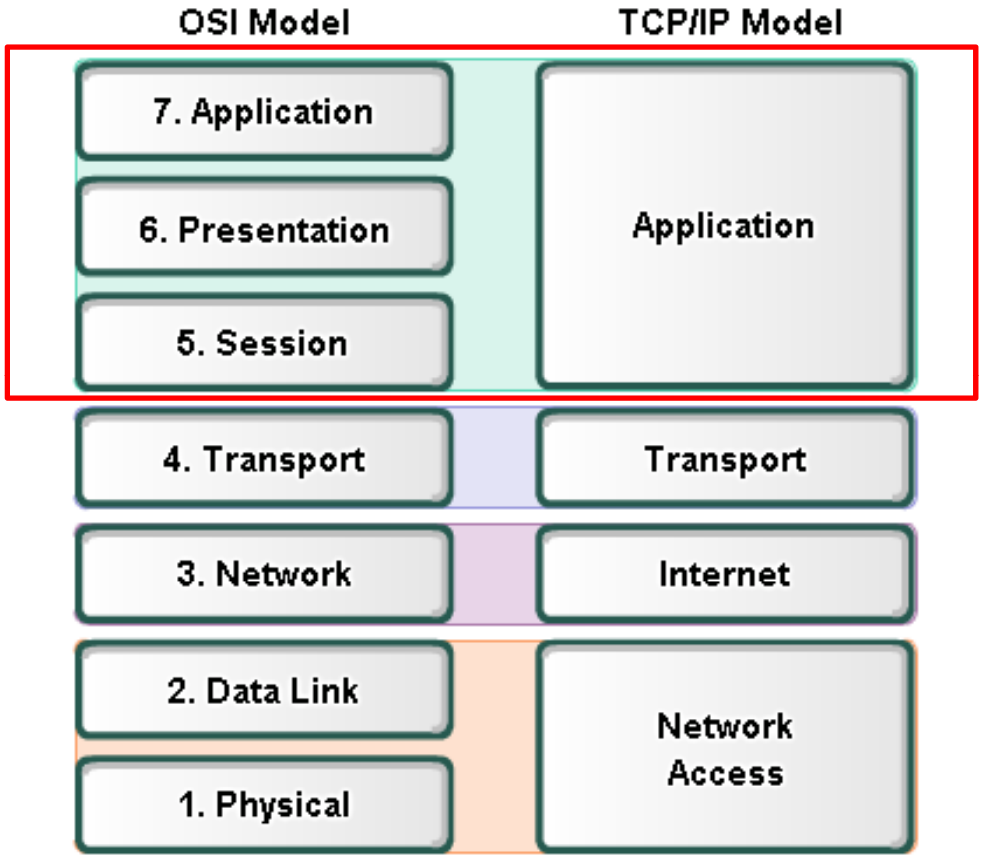
Putty into Elrond as root, make a bin directory, and create the following script in that directory. Use copy and paste and be sure and give it execute permission.

```
[root@elrond bin]# mkdir bin
[root@elrond bin]# cd bin
[root@elrond bin]# vi set-gateway
#!/bin/bash
#
# Prompt user for location and set the gateway permanently
#
echo -n "Enter 1 for classroom or 2 for lab: "
read answer
if [ "$answer" = "1" ]; then
   gw="172.30.1.1"
elif [ "$answer" = "2" ]; then
   gw="172.30.4.1"
else
   echo No changes made
   exit 1
fi
echo Setting default gateway to $gw
sed -i '/GATEWAY/ d' /etc/sysconfig/network
sed -i '$aGATEWAY='$gw'' /etc/sysconfig/network
service network restart
echo Gateway:
route -n | grep UG
[root@elrond bin]# chmod +x set-gateway
```

```
[root@elrond bin]# ./set-gateway
Enter 1 for classroom or 2 for lab: 1
Setting default gateway to 172.30.1.1
Shutting down interface eth0:                [  OK  ]
Shutting down interface eth1:                [  OK  ]
Shutting down loopback interface:            [  OK  ]
Disabling IPv4 packet forwarding:  net.ipv4.ip_forward = 0
                                             [  OK  ]
Bringing up loopback interface:              [  OK  ]
Bringing up interface eth0:                  [  OK  ]
Bringing up interface eth1:                  [  OK  ]
Gateway:
0.0.0.0         172.30.1.1      0.0.0.0         UG    0      0        0 eth0
[root@elrond bin]#
```

Application Layer

Protocol and Reference Models



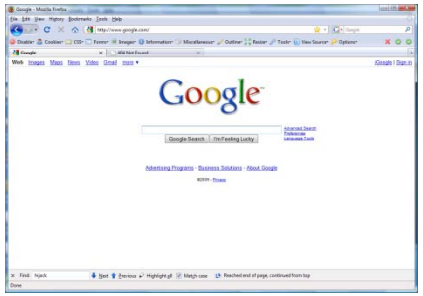
- The **Open Systems Interconnection (OSI)** model is the *most widely known internetwork reference model*.

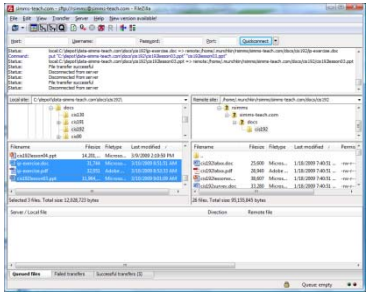
Application Layer

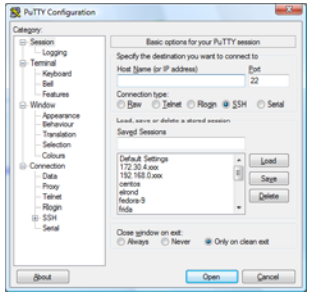
Applications

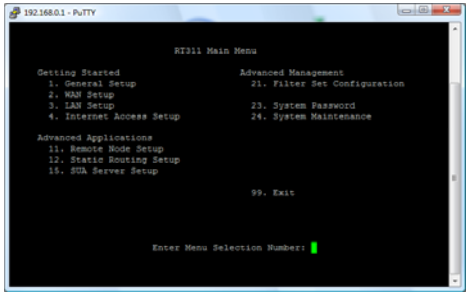
Examples:

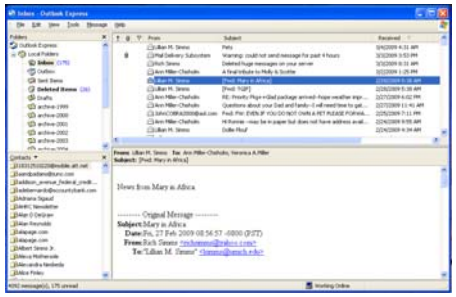
- Web servers
- FTP servers
- SSH daemon
- Telnet server
- email











Application Layer

Responsibilities of Applications

Network connections, routing, and transfer of data are all taken care of by the lower layers of the protocol stack. What must applications do?

- Authenticate users
- Control access
- Log important information
- Format data (compress/encrypt)
- Provide whatever functionality is desired.

Service Ports

Previously we talked about Layer 4 ports. Ports are used to direct requests to the appropriate service/application

```
< snipped >
# 21 is registered to ftp, but also used by fsp
ftp             21/tcp
ftp             21/udp          fsp fspd
ssh             22/tcp                          # SSH Remote Login Protocol
ssh             22/udp                          # SSH Remote Login Protocol
telnet          23/tcp
telnet          23/udp
# 24 - private mail system
lmtp            24/tcp                          # LMTP Mail Delivery
lmtp            24/udp                          # LMTP Mail Delivery
smtp            25/tcp          mail
smtp            25/udp          mail
< snipped >
domain          53/tcp                          # name-domain server
domain          53/udp
whois++         63/tcp
whois++         63/udp
bootps          67/tcp                          # BOOTP server
bootps          67/udp
bootpc          68/tcp          dhcpc           # BOOTP client
bootpc          68/udp          dhcpc
tftp            69/tcp
tftp            69/udp
finger          79/tcp
finger          79/udp
http            80/tcp          www www-http    # WorldWideWeb HTTP
http            80/udp          www www-http    # HyperText Transfer Protocol
kerberos        88/tcp          kerberos5 krb5  # Kerberos v5
< snipped >
```

Application Layer

The Client-Server Model

Clients

Programs that are generally run on demand, and initiate the network connection to the server.
Examples: telnet, ftp, ssh, browsers, email clients.

Servers

Programs (services/daemons) that are constantly running in the background waiting for client connections.

- Services and Ports: */etc/services*
- Architecture:
  - Direct or iterative servers – listens to a particular port and directly responds to requests
  - Indirect or concurrent servers (e.g. super daemons) – listens to a particular port and then starts up another server program to process the request

Application Layer

The Super Daemons

- There are three primary super-daemons controlling server services.
- Super daemons spawn other daemons to handle specific client requests.

1. inetd - From early UNIX days, this was the primary daemon for handling tcp application services. It is being replaced by xinetd.

2. portmap - portmapper operates with Remote Procedure Call (RCP) applications.

3. xinetd - Extended Internet Services Daemon: used by modern distributions of Linux.

Application Layer

Steps to installing services

1. Install software package using **yum**, **rpm** or build from source code
2. Customize service's configuration file
3. Modify the firewall to allow access to the service
4. Customize SELinux context settings to allow use
5. Start the service
6. Configure service to automatically start when system boots
7. Monitor and verify service is running
8. Troubleshoot as necessary
9. Monitor log files as appropriate
10. Configure additional security

Application Layer

Some Lingo

1. Application – generic term for a software package. It could be a client or server application.

2. Service – runs in the background on a server waiting for requests to handle.

3. Daemon – term used in the UNIX world to refer to a service.

4. Super daemon – An "umbrella" service that listens for requests then invokes the appropriate service to handle it.

Service and daemon will be used interchangeably

telnet
(via port 23)

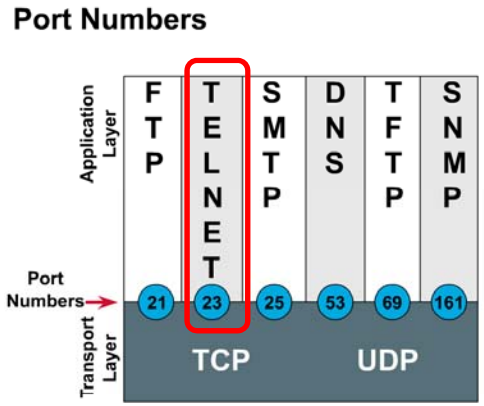
Installing and Configuring Telnet (Red Hat Family)

Telnet

- Provides command line interface to a remote host
- Client-server model
- Uses port 23
- Not secure, uses clear text over the network that can be sniffed

```
[root@elrond bin]# cat /etc/services | grep -w 23
# $Id: services,v 1.42 2006/02/23 13:09:23 pknirsch Exp $
telnet          23/tcp
telnet          23/udp
[root@elrond bin]#
```

Telnet uses port 23



Installing and Configuring Telnet (Red Hat Family)

Is it installed?

```
[root@bigserver ~]# rpm -qa | grep telnet
telnet-0.17-39.el5
telnet-server-0.17-39.el5
[root@bigserver ~]#
```

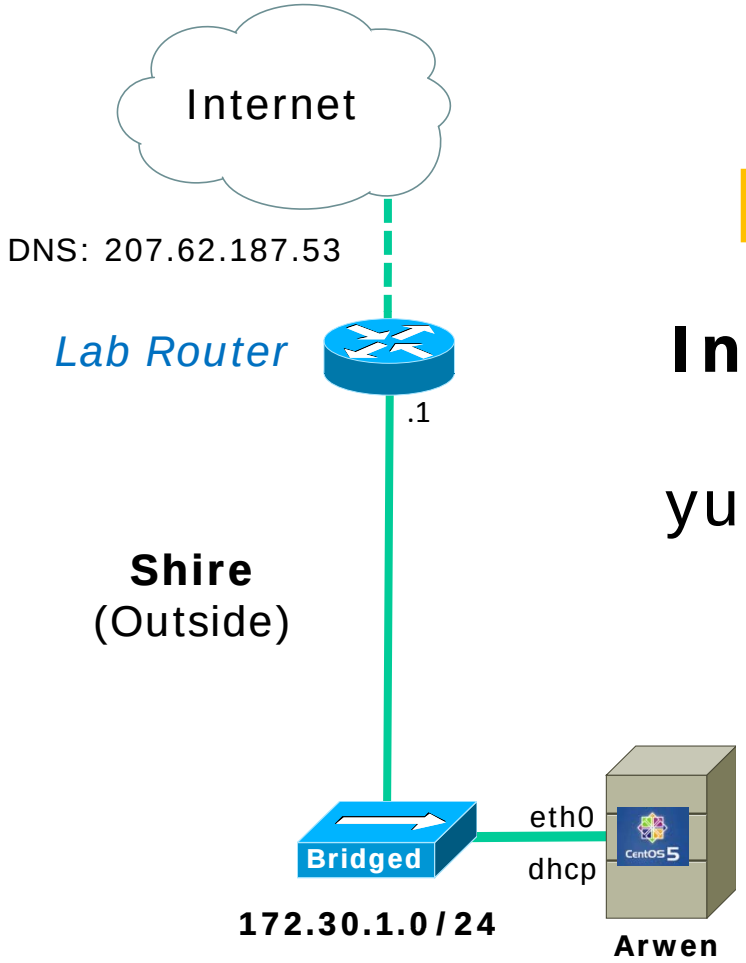
client

server

No response means it is not installed

*Use **dpkg –l | grep telnet** on the Debian family*

Installing and Configuring Telnet



Step 1 *Installing service*

Installing Telnet

yum install telnet-server

Installing and Configuring Telnet

```
[root@bigserver ~]# yum install telnet-server
Loading "fastestmirror" plugin
Loading mirror speeds from cached hostfile
 * base: centos.mirrors.redwire.net
 * updates: centos.mirrors.redwire.net
 * addons: centos.mirrors.redwire.net
 * extras: mirrors.usc.edu
Setting up Install Process
Parsing package install arguments
Resolving Dependencies
--> Running transaction check
---> Package telnet-server.i386 1:0.17-39.el5 set to be updated
--> Processing Dependency: xinetd for package: telnet-server
--> Running transaction check
---> Package xinetd.i386 2:2.3.14-10.el5 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved
```

Note that the telnet server uses xinetd

Installing and Configuring Telnet

```
Dependencies Resolved

================================================================================
 Package                 Arch       Version           Repository        Size
================================================================================
Installing:
 telnet-server           i386       1:0.17-39.el5     base               35 k
Installing for dependencies:
 xinetd                  i386       2:2.3.14-10.el5   base              124 k

Transaction Summary
================================================================================
Install      2 Package(s)
Update       0 Package(s)
Remove       0 Package(s)

Total download size: 159 k
Is this ok [y/N]: y
Downloading Packages:
(1/2): xinetd-2.3.14-10.e 100% |=========================| 124 kB    00:00
(2/2): telnet-server-0.17 100% |=========================|  35 kB    00:00
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing: xinetd                       ########################### [1/2]
  Installing: telnet-server                ########################### [2/2]

Installed: telnet-server.i386 1:0.17-39.el5
Dependency Installed: xinetd.i386 2:2.3.14-10.el5
Complete!
[root@bigserver ~]#
```

Note, that xinetd, the super daemon, is also installed because it is a dependency of the telnet server

Installing and Configuring Telnet

Step 2 *Customize the configuration file*

```
[root@arwen ~]# cat /etc/xinetd.d/telnet
# default: on
# description: The telnet server serves telnet sessions; it uses \
#       unencrypted username/password pairs for authentication.
service telnet
{
        flags           = REUSE     Sets the TCP/IP socket to be reusable (after restarts)
        socket_type     = stream
        wait            = no        Starts a daemon for each request
        user            = root      Sets UID for daemon
        server          = /usr/sbin/in.telnetd
        log_on_failure  += USERID   Use HOST for IP address and RECORD for terminal type
        disable         = no        This enables Telnet service
}
```

Great reference is "LINUX TCP/IP Network Administration" by Scott Mann

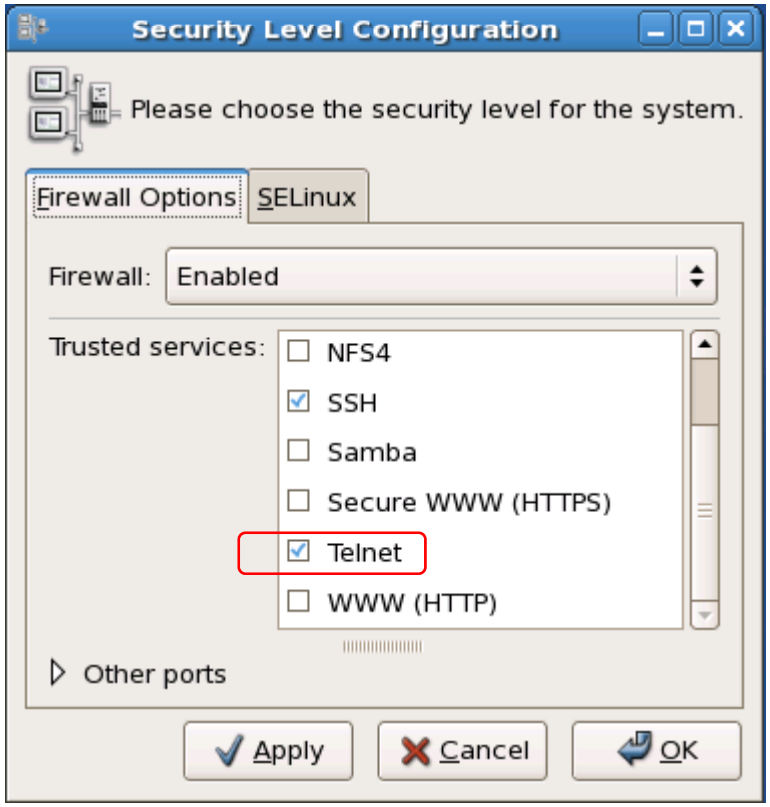
Telnet service activity

Step 2 *Modify Arwen's telnet configuration file as shown below*

Arwen

```
[root@arwen ~]# cat /etc/xinetd.d/telnet
# default: on
# description: The telnet server serves telnet sessions; it uses \
#       unencrypted username/password pairs for authentication.
service telnet
{
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/in.telnetd
        log_on_failure  += USERID
        disable         = no
}
```

Installing and Configuring Telnet (Red Hat Family)



Step 3

Modify the firewall to allow incoming new Telnet (TCP port 23) connections

This Red Hat GUI tool keeps track of its settings in ***/etc/sysconfig/system-config-securitylevel****.*

Warning: Don't configure the firewall using ***iptables*** *commands and this GUI tool. The GUI tool may clobber any settings you make using* ***iptables****.*

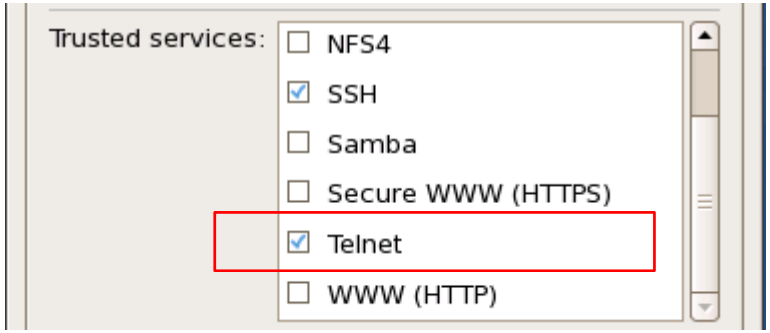
From the command line:

```
iptables -I RH-Firewall-1-INPUT 10 -m state --state NEW -m tcp -p tcp --dport 23 -j ACCEPT
iptables-save > /etc/sysconfig/iptables
```

Installing and Configuring Telnet

CentOS default firewall modified

```
root@bigserver ~]# cat /etc/sysconfig/iptables
# Firewall configuration written by system-config-securitylevel
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp --icmp-type any -j ACCEPT
-A RH-Firewall-1-INPUT -p 50 -j ACCEPT
-A RH-Firewall-1-INPUT -p 51 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp --dport 5353 -d 224.0.0.251 -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 ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 23 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
[root@bigserver ~]#
```



Telnet (23) port is now open for incoming ***new*** *connections (a "trusted service")*

Firewall for Telnet

CentOS Modified

Current firewall settings

```
[root@arwen ~]# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0           icmp type 255
ACCEPT     esp  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     ah   --  0.0.0.0/0            0.0.0.0/0
ACCEPT     udp  --  0.0.0.0/0            224.0.0.251         udp dpt:5353
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:631
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:631
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:23
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-prohibited
[root@arwen ~]#
```

Telnet port is now open

*Note: **iptables –L –n** shows current firewall settings, **cat /etc/sysconfig/iptables** shows firewall and NAT settings that will be configured at system boot*

Telnet service activity

Step 3 *Modify Arwen's firewall to allow telnet connections*

Arwen

Add new rule

```
[root@arwen ~]# iptables -I RH-Firewall-1-INPUT 10 -m state --state NEW -m tcp -p tcp --dport 23 -j ACCEPT
```

Verify rule was added to current firewall

```
[root@arwen ~]# iptables -n -L | grep 23
ACCEPT     tcp  --  0.0.0.0/0           0.0.0.0/0           state NEW tcp dpt:23
```

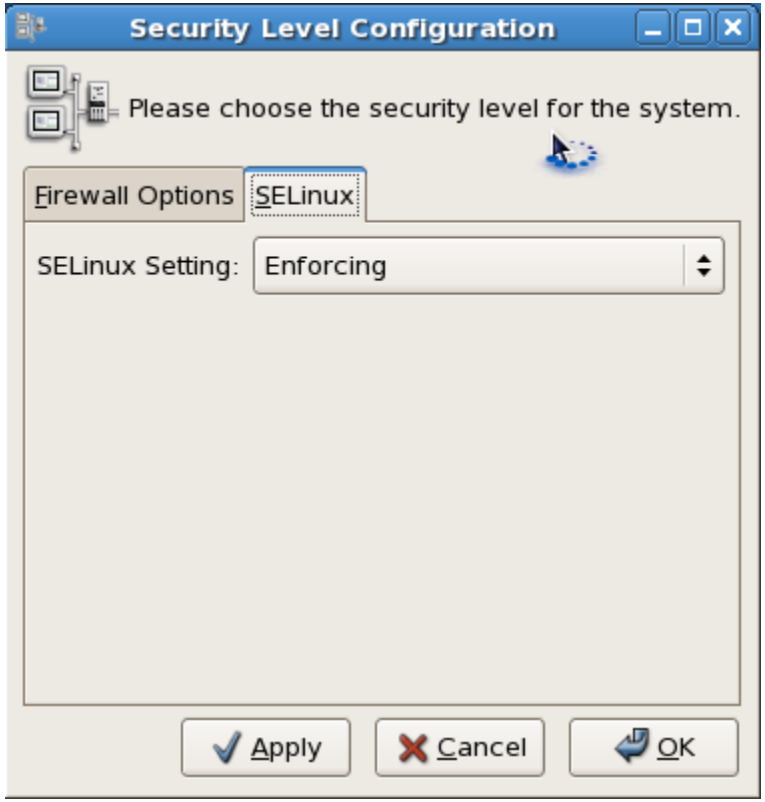
Make firewall permanent

```
[root@arwen ~]# iptables-save > /etc/sysconfig/iptables
```

Show permanent firewall

```
[root@arwen ~]# cat /etc/sysconfig/iptables
# Generated by iptables-save v1.3.5 on Sun Mar 14 20:46:15 2010
< snipped >
-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 tcp -m state --state NEW -m tcp --dport 23 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Sun Mar 14 20:46:15 2010
[root@arwen ~]#
```

Installing and Configuring Telnet (Red Hat Family)



Step 4

Leave as Enforcing.

No changes are needed.

Installing and Configuring Telnet (Red Hat Family)

Step 5 *Start service*

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

Note telnet runs under the superdaemon xinetd umbrella

Telnet service activity

Step 5 *Start up service*

Arwen

Start telenet service by starting xinetd super daemon

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

Installing and Configuring Telnet

If service is already running use the following to reread configuration files:

service xinetd restart

or

killall -1 xinetd

hangup signal

Installing and Configuring Telnet (Red Hat Family)

Step 6

To automatically start service at system boot use:

```
[root@arwen ~]# chkconfig xinetd on
[root@arwen ~]# chkconfig --list xinetd
xinetd          0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@arwen ~]#
```

To not start service at system boot use:

```
[root@arwen ~]# chkconfig xinetd off
[root@arwen ~]# chkconfig --list xinetd
xinetd          0:off   1:off   2:off   3:off   4:off   5:off   6:off
[root@arwen ~]#
```

Note telnet runs under the superdaemon xinetd umbrella

Installing and Configuring Telnet

```
[root@arwen ~]# chkconfig –list

< snipped >

xinetd based services:
        chargen-dgram:  off
        chargen-stream: off
        daytime-dgram:  off
        daytime-stream: off
        discard-dgram:  off
        discard-stream: off
        echo-dgram:     off
        echo-stream:    off
        eklogin:        off
        ekrb5-telnet:   off
        gssftp:         off
        klogin:         off
        krb5-telnet:    off
        kshell:         off
        rsync:          off
        tcpmux-server:  off
        telnet:         on
        time-dgram:     off
        time-stream:    off
```

xinetd is a super daemon which acts as an umbrella for many other services

Multiple telnet daemons

Telnet service activity

Step 6 *Configure automatic service startup*

Arwen

Automatically start up telnet (via xinetd super daemon) at system boot

```
[root@arwen ~]# chkconfig xinetd on
[root@arwen ~]# chkconfig --list xinetd
xinetd          0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@arwen ~]#
```

Verify telnet is enabled (set to “on”)

```
[root@arwen ~]#  chkconfig --list | grep telnet
        ekrb5-telnet:   off
        krb5-telnet:    off
        telnet:         on
```

Installing and Configuring Telnet

Step 7 *Monitor and verify service is running*

telnetd processes

```
[cis192@bigserver ~]$ ps -ef | grep telnet
root      6156  6118  0 07:52 ?        00:00:00 in.telnetd: kate
root      6268  6118  0 07:53 ?        00:00:00 in.telnetd: 192.168.0.27
root      6299  6118  0 07:56 ?        00:00:00 in.telnetd: 192.168.0.23
cis192    6325  6270  0 07:56 pts/2    00:00:00 grep telnet
[cis192@bigserver ~]$
```

Individual telnetd daemons are run for each session

Installing and Configuring Telnet

Step 7 *Verify service is running*

netstat

```
[root@bigserver ~]# netstat -tl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 r1.localdomain:2208        *:*                         LISTEN
tcp        0      0 *:sunrpc                   *:*                         LISTEN
tcp        0      0 *:x11                      *:*                         LISTEN
tcp        0      0 *:ftp                      *:*                         LISTEN
tcp        0      0 *:telnet                   *:*                         LISTEN
tcp        0      0 r1.localdomain:ipp         *:*                         LISTEN
tcp        0      0 *:792                      *:*                         LISTEN
tcp        0      0 r1.localdomain:smtp        *:*                         LISTEN
tcp        0      0 r1.localdomain:2207        *:*                         LISTEN
tcp        0      0 *:x11                      *:*                         LISTEN
tcp        0      0 *:ssh                      *:*                         LISTEN
[root@bigserver ~]#
```

*Use **netstat –tl** command to see what port names your system is listening for requests on*

Installing and Configuring Telnet

Step 7 *Verify service is running*

netstat

```
[root@bigserver ~]# netstat -tln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 127.0.0.1:2208              0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:111                 0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:6000                0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:21                  0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:23                  0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:631               0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:792                 0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:25                0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:2207              0.0.0.0:*                   LISTEN
tcp        0      0 :::6000                     :::*                        LISTEN
tcp        0      0 :::22                       :::*                        LISTEN
[root@bigserver ~]#
```

*Use **netstat -tln** command to see what port numbers your system is listening for requests on*

Telnet service activity

Step 8 *Verify service is running*

netstat

```
[root@bigserver ~]# netstat -tln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 127.0.0.1:2208              0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:111                 0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:6000                0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:21                  0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:23                  0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:631               0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:792                 0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:25                0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:2207              0.0.0.0:*                   LISTEN
tcp        0      0 :::6000                     :::*                        LISTEN
tcp        0      0 :::22                       :::*                        LISTEN
[root@bigserver ~]#
```

*Use **netstat -tln** command to see what port numbers your system is listening for requests on*

Installing and Configuring Telnet

Try it! *From Elrond we log in to Arwen using Telnet*

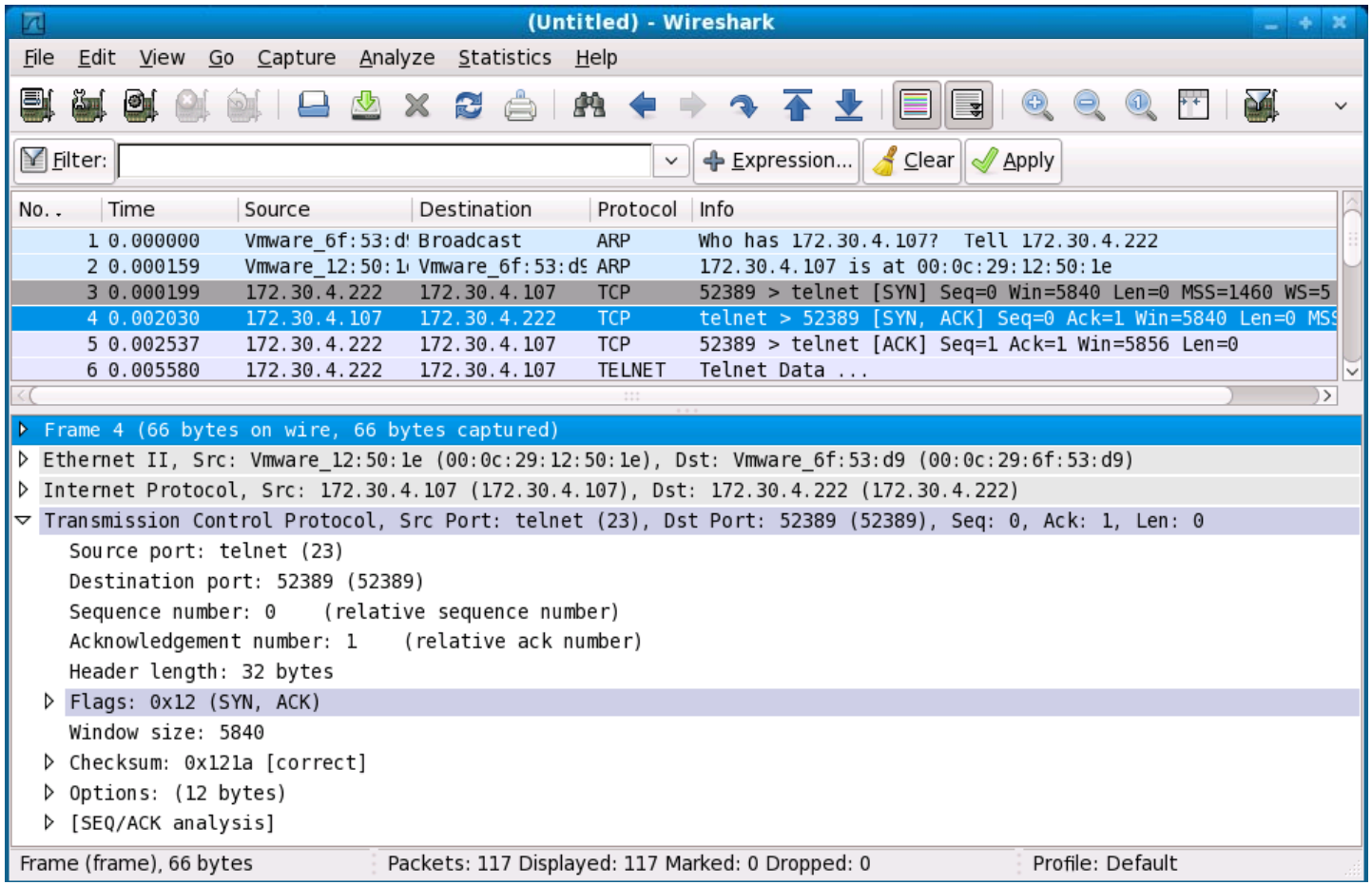
```
[root@elrond ~]# telnet 192.168.2.9
Trying 192.168.2.9...
Connected to arwen (192.168.2.9).
Escape character is '^]'.
CentOS release 5.4 (Final)
Kernel 2.6.18-164.el5 on an i686
login: cis192
Password:
Last login: Tue Mar 16 02:49:57 from 172.30.1.155
[cis192@arwen ~]$ echo success!
success!
[cis192@arwen ~]$ exit
logout
Connection closed by foreign host.
[root@elrond ~]#
```

Telnet service activity

Try it! *From Elrond we log in to Arwen using Telnet*

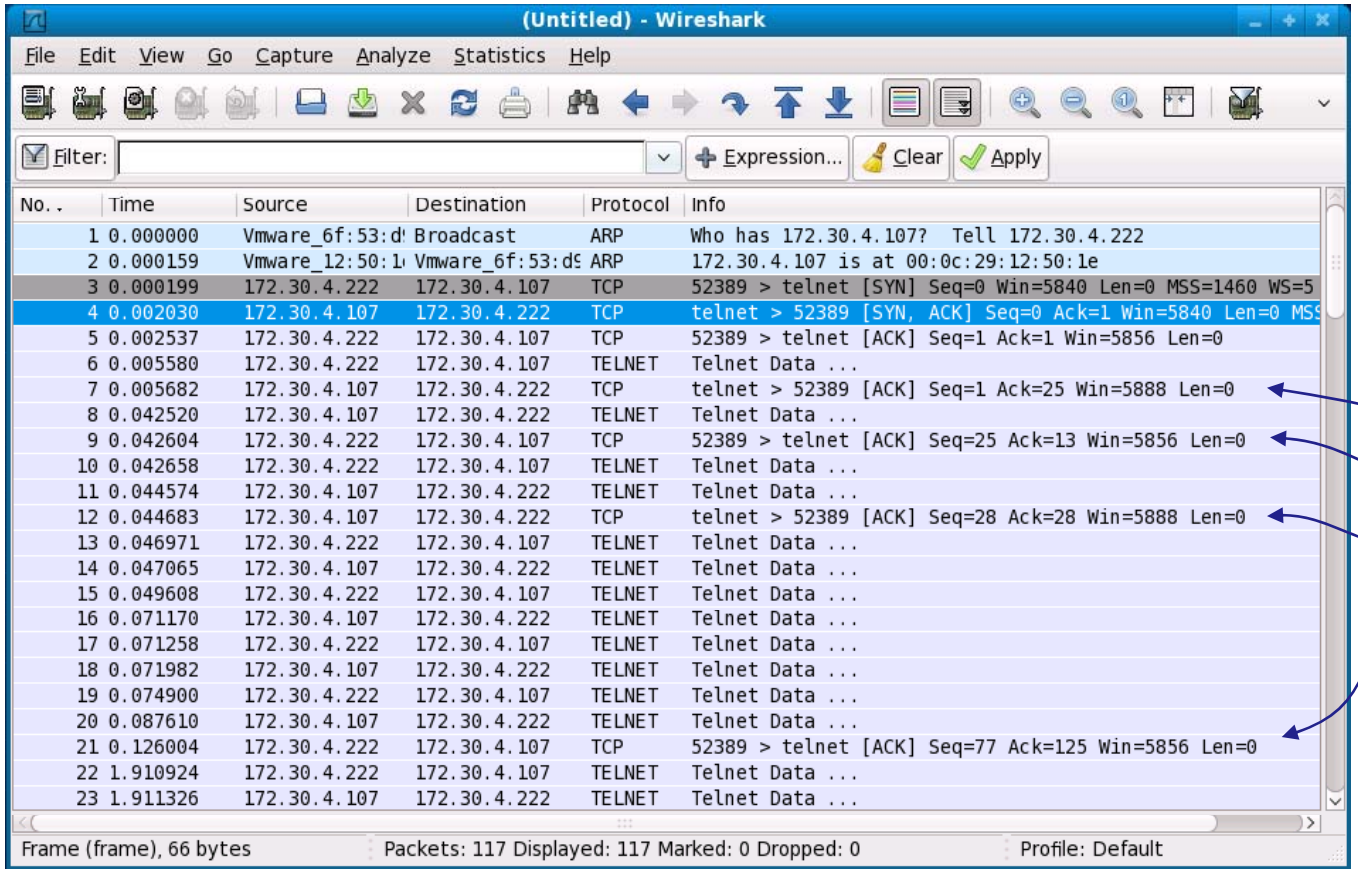
```
[root@elrond ~]# telnet 192.168.2.9
Trying 192.168.2.9...
Connected to arwen (192.168.2.9).
Escape character is '^]'.
CentOS release 5.4 (Final)
Kernel 2.6.18-164.el5 on an i686
login: cis192
Password:
Last login: Tue Mar 16 02:49:57 from 172.30.1.155
[cis192@arwen ~]$ echo success!
success!
[cis192@arwen ~]$ exit
logout
Connection closed by foreign host.
[root@elrond ~]#
```

Installing and Configuring Telnet



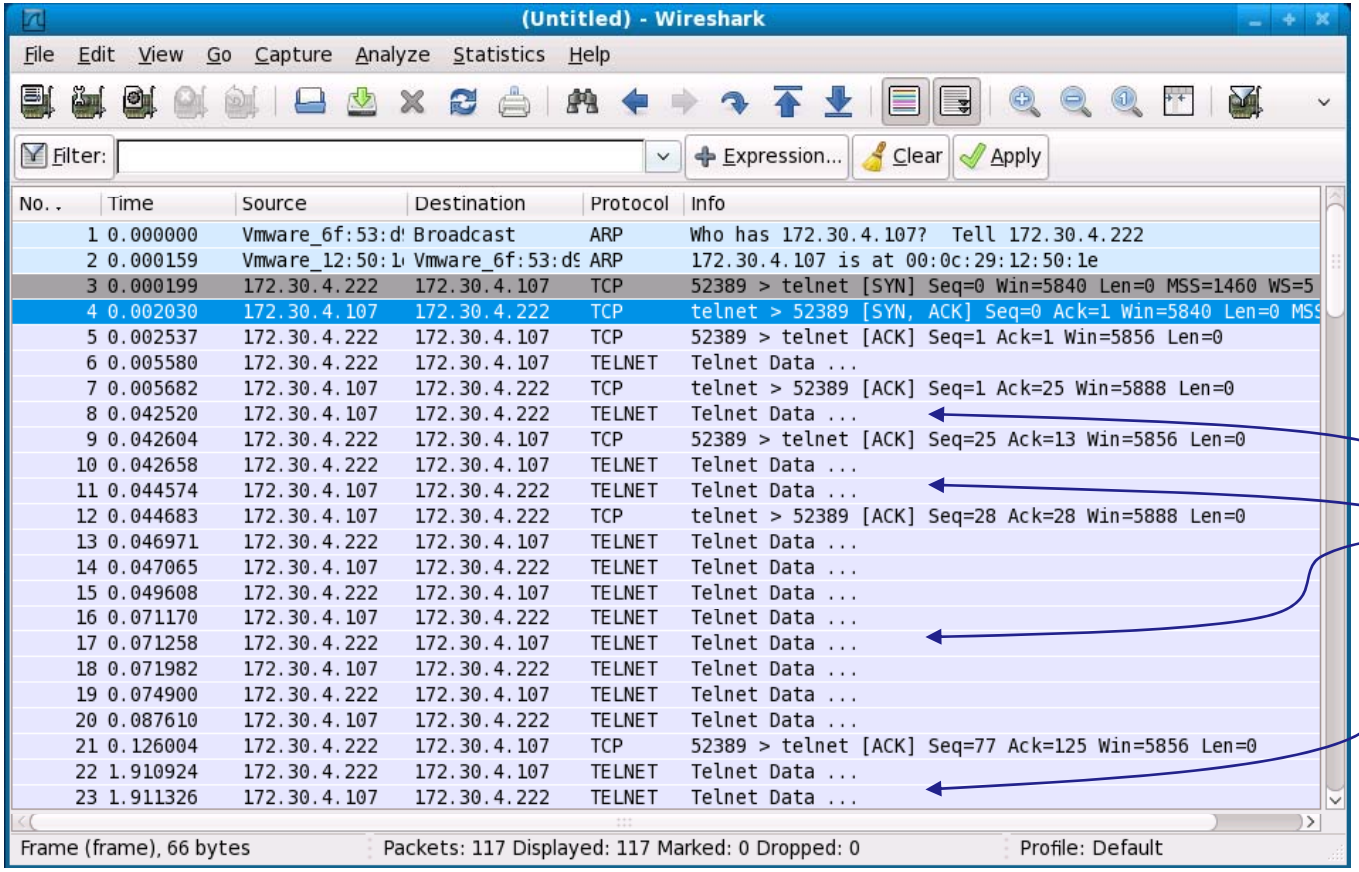
3-way handshake that initiates TCP connection

Installing and Configuring Telnet



TCP acknowledgments (ACKS) of data received

Installing and Configuring Telnet



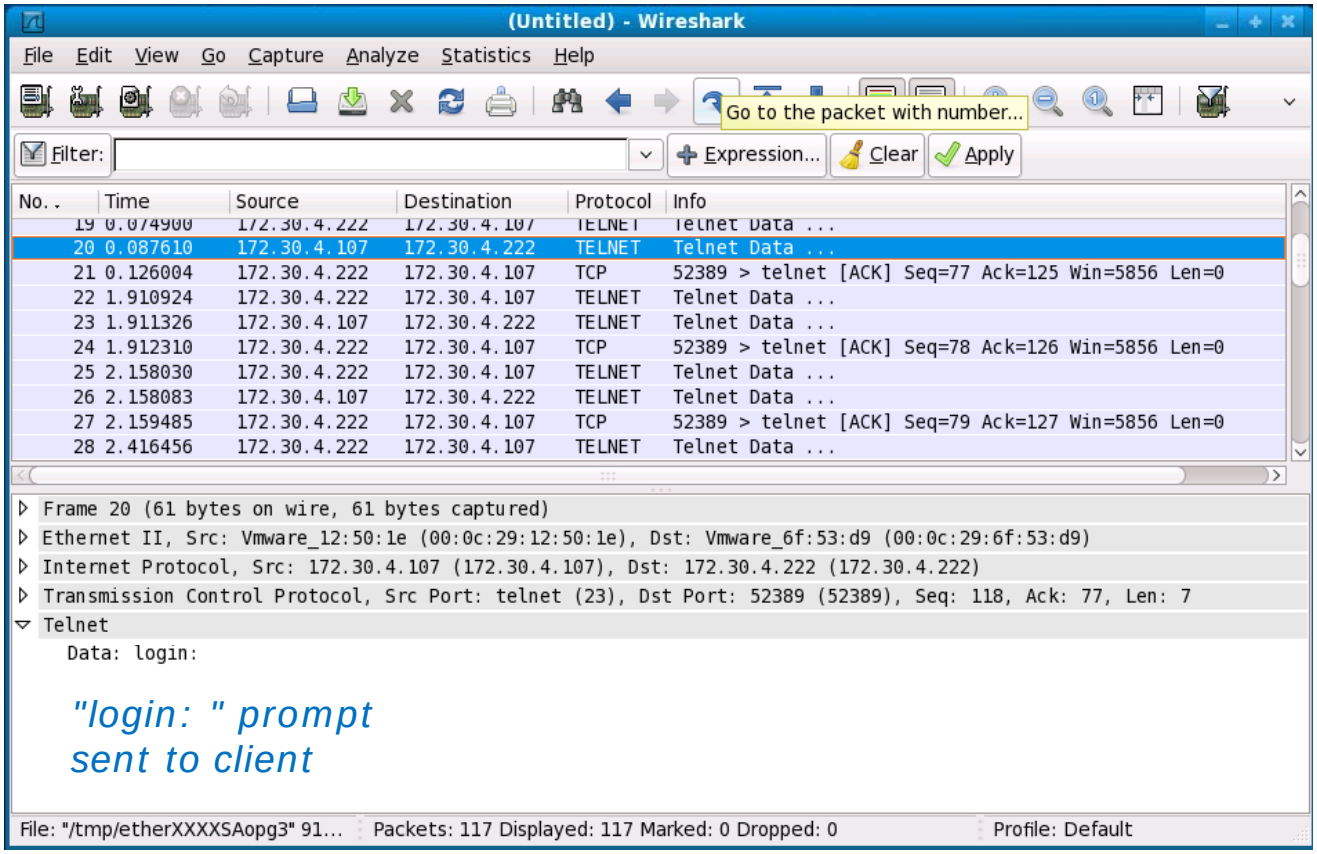
Installing and Configuring Telnet

Layer 2 Link frame with MAC addresses

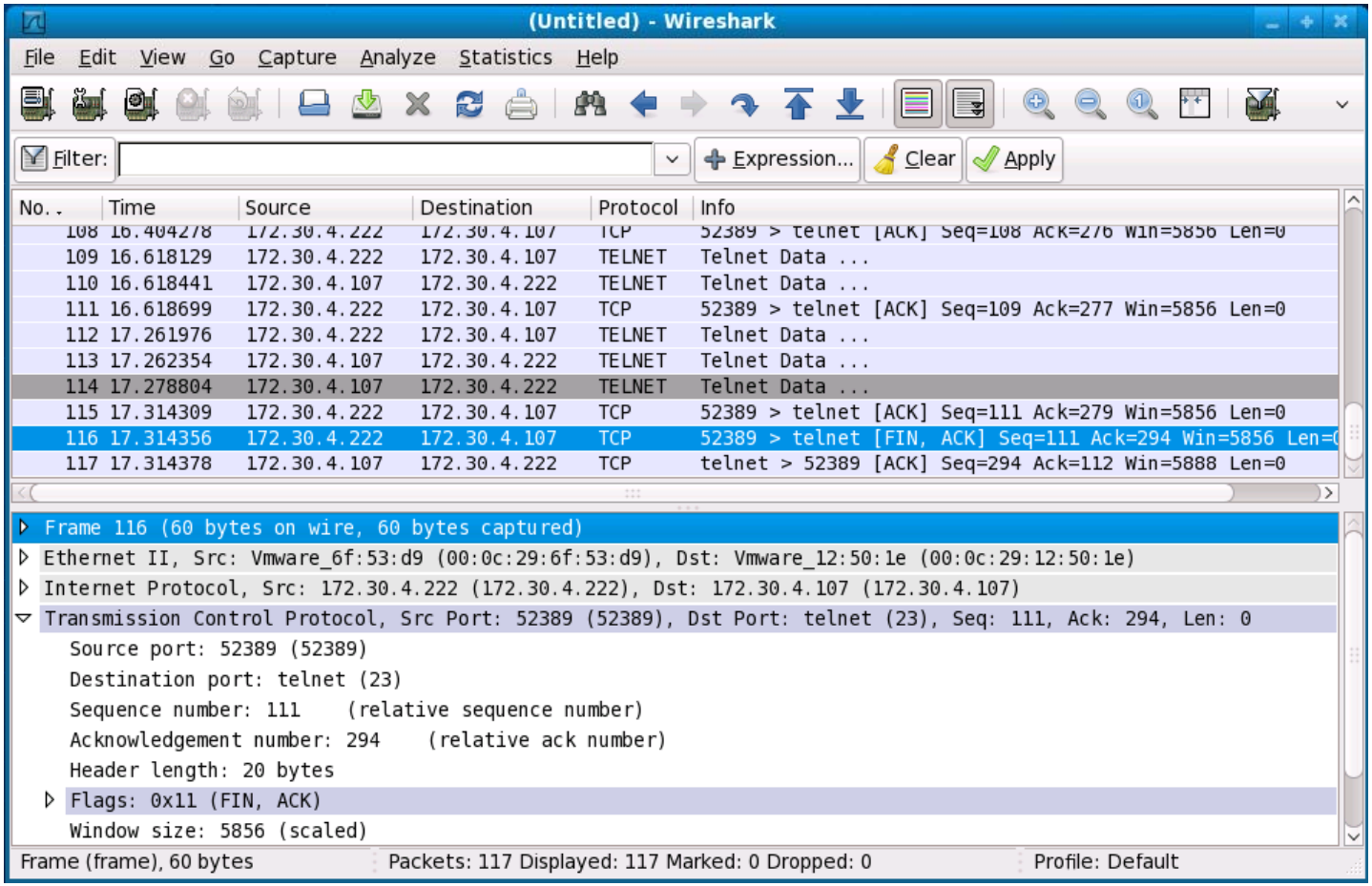
Layer 3 Network packet with IP addresses

Layer 4 Transport segment with port numbers

Layer 5 Application data



Installing and Configuring Telnet



Connection terminated

Installing and Configuring Telnet

Step 8 Troubleshooting

```
root@sun:~# telnet 172.30.4.107
Trying 172.30.4.107...
telnet: Unable to connect to remote host: Connection refused
```

Open the firewall on the Telnet sever to accept incoming Telnet connections

```
root@sun:~# telnet 172.30.4.107
Trying 172.30.4.107...
Connected to 172.30.4.107.
Escape character is '^]'.
getaddrinfo: localhost Name or service not known
Connection closed by foreign host.
```

On the Telnet server, insure that the server's own hostname is configured in /etc/hosts

Installing and Configuring Telnet

Troubleshooting (continued)

```
root@kate:~# telnet 172.30.1.107
Trying 172.30.1.107...
Connected to 172.30.1.107.
Escape character is '^]'.
Connection closed by foreign host.
root@kate:~# telnet 172.30.1.107
```

Check:

1. *only_from and no_access attributes in* ***/etc/xinetd.d/telnet***
2. *TCP wrappers files* ***/etc/hosts.allow*** *and* ***/etc/hosts.deny***

Installing and Configuring Telnet

Troubleshooting (continued)

```
[root@elrond ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.2.8     0.0.0.0         255.255.255.252 U     0      0        0 eth1
172.30.4.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     0      0        0 eth1
0.0.0.0         172.30.4.1      0.0.0.0         UG    0      0        0 eth0
[root@elrond ~]# ping -c1 arwen
PING arwen (192.168.2.9) 56(84) bytes of data.
64 bytes from arwen (192.168.2.9): icmp_seq=1 ttl=64 time=0.040 ms

--- arwen ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.040/0.040/0.040/0.000 ms
[root@elrond ~]# telnet arwen
Trying 192.168.2.9...
telnet: connect to address 192.168.2.9: No route to host
telnet: Unable to connect to remote host: No route to host
[root@elrond ~]#
```

Opening the firewall on the Telnet sever to accept incoming Telnet connections fixed this. These wasn't any routing issue even though that's what the error message indicated!

Installing and Configuring Telnet

Step 9 **Monitor log files**

```
[root@arwen ~]# cat /var/log/messages | grep telnet
Mar 14 00:00:38 arwen yum: Installed: 1:telnet-server-0.17-39.el5.i386
Mar 14 01:02:29 arwen xinetd[2108]: readjusting service telnet
Mar 14 01:32:15 arwen xinetd[2108]: START: telnet pid=12158 from=192.168.2.10
Mar 14 01:33:15 arwen xinetd[2108]: EXIT: telnet status=0 pid=12158
duration=60(sec)
Mar 14 08:33:05 arwen xinetd[2108]: START: telnet pid=16867 from=172.30.4.159
Mar 14 08:33:27 arwen xinetd[2108]: EXIT: telnet status=0 pid=16867
duration=22(sec)
Mar 14 09:07:19 arwen xinetd[2108]: START: telnet pid=16954 from=172.30.4.159
Mar 14 09:07:38 arwen xinetd[2108]: EXIT: telnet status=0 pid=16954
duration=19(sec)
< snipped >
Mar 16 03:06:51 arwen xinetd[2192]: START: telnet pid=2611 from=172.30.1.155
Mar 16 03:06:56 arwen xinetd[2192]: EXIT: telnet status=0 pid=2611
duration=5(sec)
[root@arwen ~]#
```

Installing and Configuring Telnet

Step 10 *Configure additional security*

```
[root@arwen ~]# cat /etc/xinetd.d/telnet
# default: on
# description: The telnet server serves telnet sessions; it uses \
#       unencrypted username/password pairs for authentication.
service telnet
{
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        only_from       = 192.168.0.23
        server          = /usr/sbin/in.telnetd
        log_on_failure  += USERID
        disable         = no
}
[root@arwen ~]#
```

Use only_from to restrict clients that can access the Telnet service

Installing and Configuring Telnet

Only_ from examples

```
only_from = arwen
```
hostname

```
only_from = arwen legolas
```
multiple hostnames

```
only_from = 192.168.3.12 192.168.3.14
```
or IP addresses

```
only_from = 192.168.3.{12, 14}
```
same as above

```
only_from = 192.168.0.0
```
0's are wildcards

```
only_from = sauron 172.30.4.0 10.10.10.{1, 200}
```
mixes

Installing and Configuring Telnet

Step 10 *Configure additional security with TCP wrappers*

TCP Wrappers

```
[root@arwen ~]# type xinetd
xinetd is /usr/sbin/xinetd
[root@arwen ~]# ldd /usr/sbin/xinetd
        linux-gate.so.1 =>  (0x00427000)
        libselinux.so.1 => /lib/libselinux.so.1 (0x004fd000)
        libwrap.so.0 => /usr/lib/libwrap.so.0 (0x00d94000)
        libnsl.so.1 => /lib/libnsl.so.1 (0x00729000)
        libm.so.6 => /lib/libm.so.6 (0x00789000)
        libcrypt.so.1 => /lib/libcrypt.so.1 (0x00cba000)
        libc.so.6 => /lib/libc.so.6 (0x00110000)
        libdl.so.2 => /lib/libdl.so.2 (0x00a6d000)
        libsepol.so.1 => /lib/libsepol.so.1 (0x002c5000)
        /lib/ld-linux.so.2 (0x00979000)
[root@arwen ~]#
```

xinetd, which invokes telnet, is compiled with TCP wrappers

- Use **/etc/hosts.allow** for permitted hosts
- Use **/etc/hosts.deny** to ban hosts

Installing and Configuring Telnet

TCP Wrappers

Arwen


```
[root@arwen ~]# cat /etc/hosts.allow
sshd: frodo 192.168. 10.0.0.0/255.0.0.0
in.telnetd:  192.168.2.1  127.0.0.1
vsvtpd: frodo arwen sauron
```

Only Elrond and local telnet access is allowed into Arwen

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

Everyone else is denied (this includes Nosmo)

Installing and Configuring Telnet

```
[root@arwen ~]# cat /etc/hosts.allow
sshd: frodo 192.168. 10.0.0.0/255.0.0.0
in.telnetd:  192.168.2.1  127.0.0.1
vsftpd: frodo arwen sauron

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

Arwen

Elrond
192.168.2.1

Access allowed

```
[root@elrond ~]# telnet arwen
Trying 192.168.2.9...
Connected to arwen (192.168.2.9).
Escape character is '^]'.
CentOS release 5.2 (Final)
Kernel 2.6.18-92.1.22.el5 on an i686
login: cis192
Password:
Last login: Mon Mar 16 00:03:58 from arwen
[cis192@arwen ~]$
```

Sauron
192.168.2.200

Access denied

```
root@sauron:~# telnet arwen
Trying 192.168.2.9...
Connected to arwen.
Escape character is '^]'.
Connection closed by foreign host.
```

vsftpd

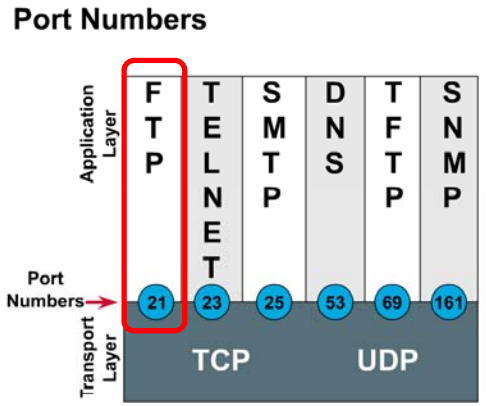
Installing and Configuring Telnet (Red Hat Family)

FTP

- File transfer protocol
- Client-server model
- Uses port 20 (for data) and 21 (for commands)
- Not secure, uses clear text over the network that can be sniffed

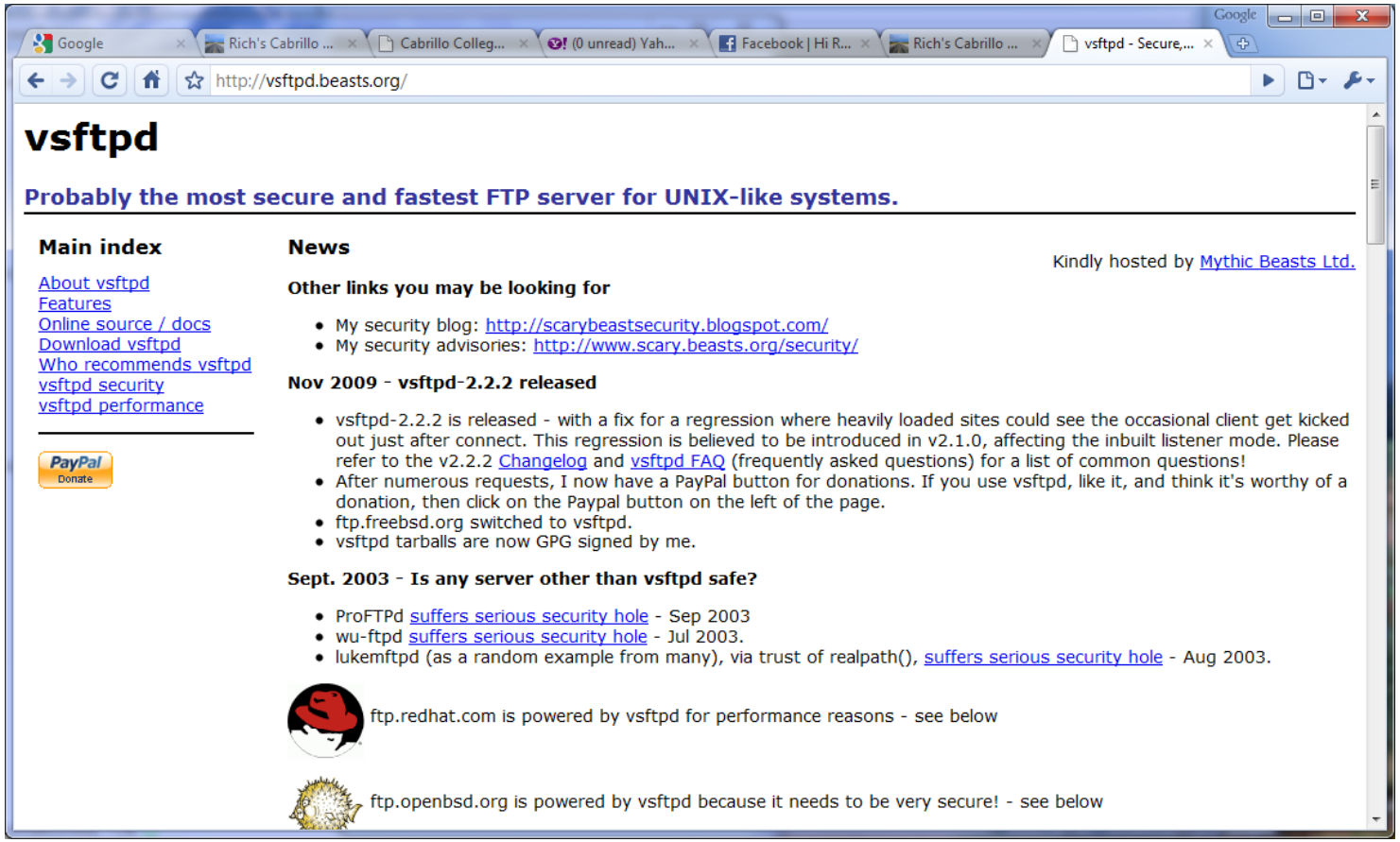
FTP uses ports 20 and 21

```
[root@elrond bin]# cat /etc/services
< snipped >
ftp-data        20/tcp
ftp-data        20/udp
# 21 is registered to ftp, but also used by fsp
ftp             21/tcp
ftp             21/udp          fsp fspd
< snipped >
[root@elrond bin]#
```



vsftpd

- vsftpd = Very Secure FTP Daemon
- Licensed under the GNU General Public License
- http://vsftpd.beasts.org/



Installing and Configuring vsftpd (Red Hat Family)

Is it installed?

```
[root@bigserver ~]# rpm -qa | grep vsftpd
vsftpd-2.0.5-12.el5
```

No response means it is not installed

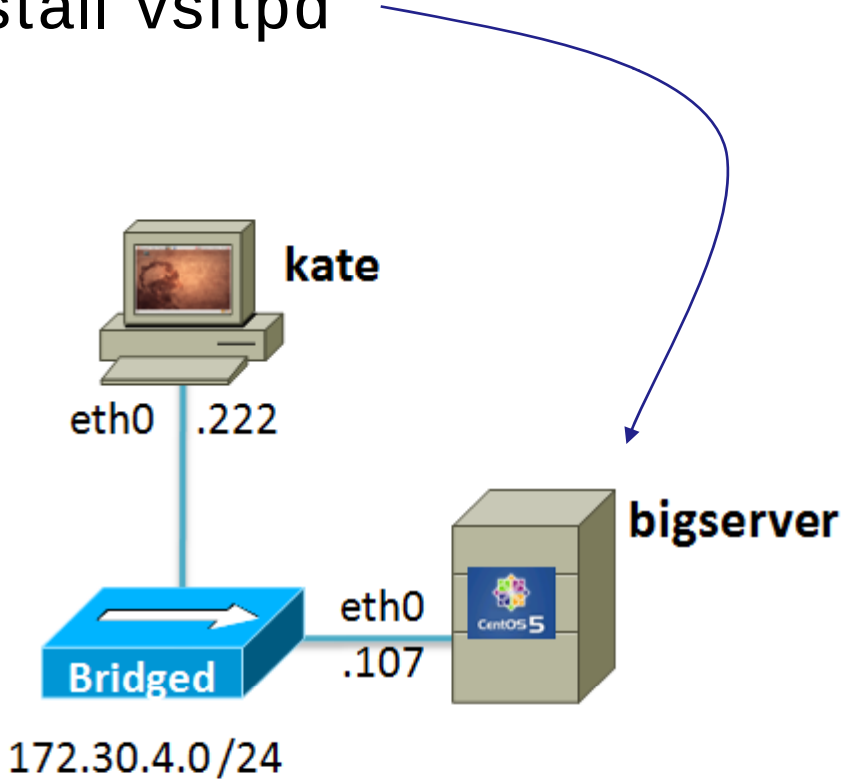
*Use **dpkg –l | grep vsftpd** on the Debian family*

vsftpd

Installing vsftpd

Step 1 *Installing service*

yum install vsftpd



vsftpd

```
[root@bigserver ~]# yum install vsftpd
Loading "fastestmirror" plugin
Loading mirror speeds from cached hostfile
 * base: mirror.hmc.edu
 * updates: mirrors.easynews.com
 * addons: mirrors.cat.pdx.edu
 * extras: centos.cogentcloud.com
Setting up Install Process
Parsing package install arguments
Resolving Dependencies
--> Running transaction check
---> Package vsftpd.i386 0:2.0.5-12.el5 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved
```

vsftpd

```
Dependencies Resolved

================================================================================
 Package                 Arch       Version          Repository        Size
================================================================================
Installing:
 vsftpd                  i386       2.0.5-12.el5     base              137 k

Transaction Summary
================================================================================
Install      1 Package(s)
Update       0 Package(s)
Remove       0 Package(s)

Total download size: 137 k
Is this ok [y/N]: y
Downloading Packages:
(1/1): vsftpd-2.0.5-12.el 100% |=========================| 137 kB    00:00
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing: vsftpd                       ######################### [1/1]

Installed: vsftpd.i386 0:2.0.5-12.el5
Complete!
[root@bigserver ~]#
```

Installing and Configuring vsftpd

Step 2 *Customize the configuration file*

```
[root@arwen ~]# cat /etc/vsftpd/vsftpd.conf
[root@bigserver ~]# cat /etc/vsftpd/vsftpd.conf
# Example config file /etc/vsftpd/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.

< snipped >

# You may fully customise the login banner string:
ftpd_banner=Welcome to the Simms FTP service.

< snipped >

tcp_wrappers=YES
[root@bigserver ~]#
```

Make your custom banner message here

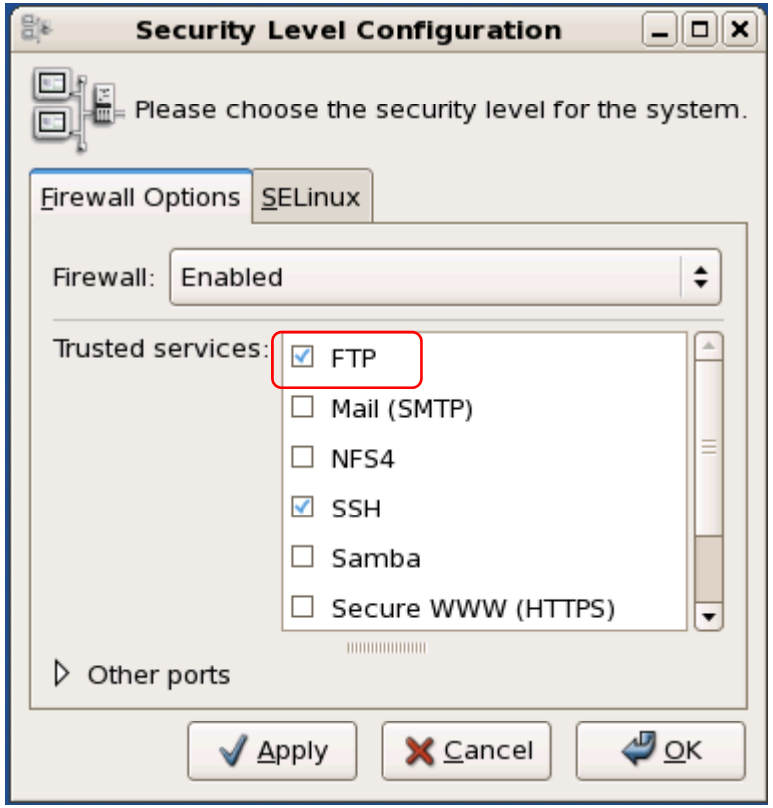
Activity

Customize the vsftpd configuration file on Arwen


```
[root@arwen ~]# cat /etc/vsftpd/vsftpd.conf
[root@bigserver ~]# cat /etc/vsftpd/vsftpd.conf

< snipped >

# You may fully customise the login banner string:
ftpd_banner=Welcome to the Simms FTP service.

< snipped >

[root@bigserver ~]#
```

Make your custom banner message here

Installing and Configuring vsftpd



Step 3

Modify the firewall to allow incoming new FTP (TCP port 21) connections.

*Note: This also causes the **ip_conntrack_ftp** module to be loaded to handle related data transfer connections.*

*This Red Hat GUI tool keeps track of its settings in: **/etc/sysconfig/system-config-securitylevel**.*

*Warning: Don't configure the firewall using **iptables** commands and this GUI tool. The GUI tool may clobber any settings you make using **iptables**.*

From the command line:

```
iptables -I RH-Firewall-1-INPUT 9 -m state --state NEW -m tcp -p tcp --dport 21 -j ACCEPT
modprobe ip_conntrack_ftp
```

Installing and Configuring vsftpd

ip_conntrack_ftp *is a kernel module. It is used to track related FTP connections so they can get through the firewall.*

From the command line (temporary)

```
[root@arwen ~]# modprobe ip_conntrack_ftp
[root@arwen ~]# lsmod | grep ftp
ip_conntrack_ftp       11569  0
ip_conntrack           53281  3 ip_conntrack_ftp,ip_conntrack_netbios_ns,xt_state
[root@arwen ~]#
```

To load at system boot (permanent), edit this file to include:

```
[root@arwen ~]# cat /etc/sysconfig/iptables-config
# Load additional iptables modules (nat helpers)
#   Default: -none-
# Space separated list of nat helpers (e.g. 'ip_nat_ftp ip_nat_irc'), which
# are loaded after the firewall rules are applied. Options for the helpers are
# stored in /etc/modprobe.conf.
IPTABLES_MODULES="ip_conntrack_netbios_ns    ip_conntrack_ftp"
< snipped >
```

Firewall for FTP

CentOS Modified

```
root@arwen ~]# cat /etc/sysconfig/iptables
# Firewall configuration written by system-config-securitylevel
# Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp --icmp-type any -j ACCEPT
-A RH-Firewall-1-INPUT -p 50 -j ACCEPT
-A RH-Firewall-1-INPUT -p 51 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp --dport 5353 -d 224.0.0.251 -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 ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 21 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 23 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
[root@arwen ~]#
[root@arwen ~]# lsmod | grep ftp
ip_conntrack_ftp        11569  0
ip_conntrack            53281  3 ip_conntrack_ftp,ip_conntrack_netbios_ns,xt_state
[root@arwen ~]#
```

Permanent firewall settings

FTP port is now open

Module to track related FTP connections is loaded

Firewall for FTP

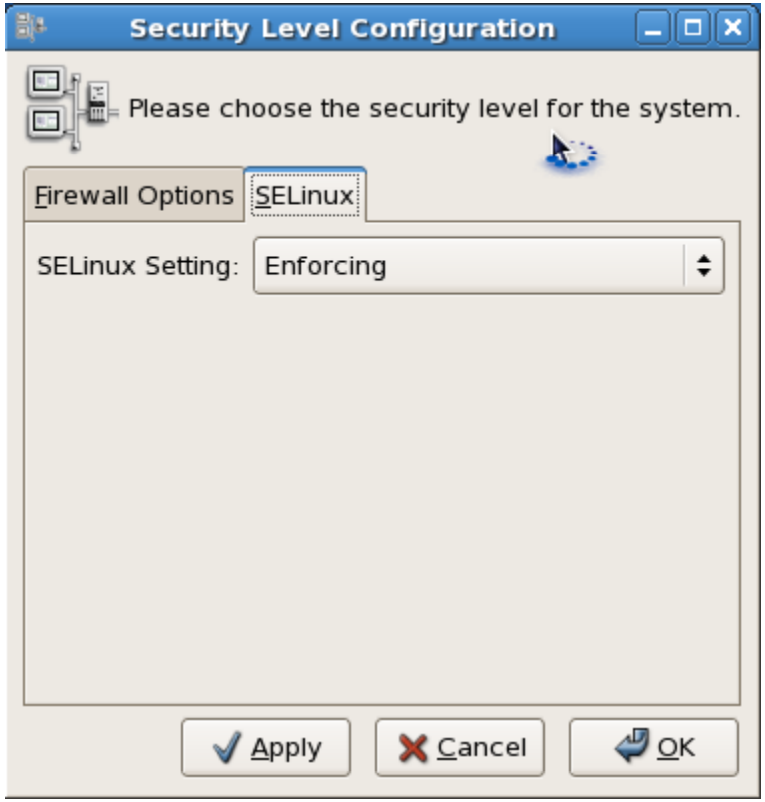
CentOS Modified

Current firewall settings

```
[root@arwen ~]# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0           0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0           0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0           icmp type 255
ACCEPT     esp  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     ah   --  0.0.0.0/0            0.0.0.0/0
ACCEPT     udp  --  0.0.0.0/0            224.0.0.251         udp dpt:5353
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:631
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:631
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:21
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:23
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-prohibited
[root@arwen ~]#
```

FTP port is now open

SELinux for FTP (CentOS)

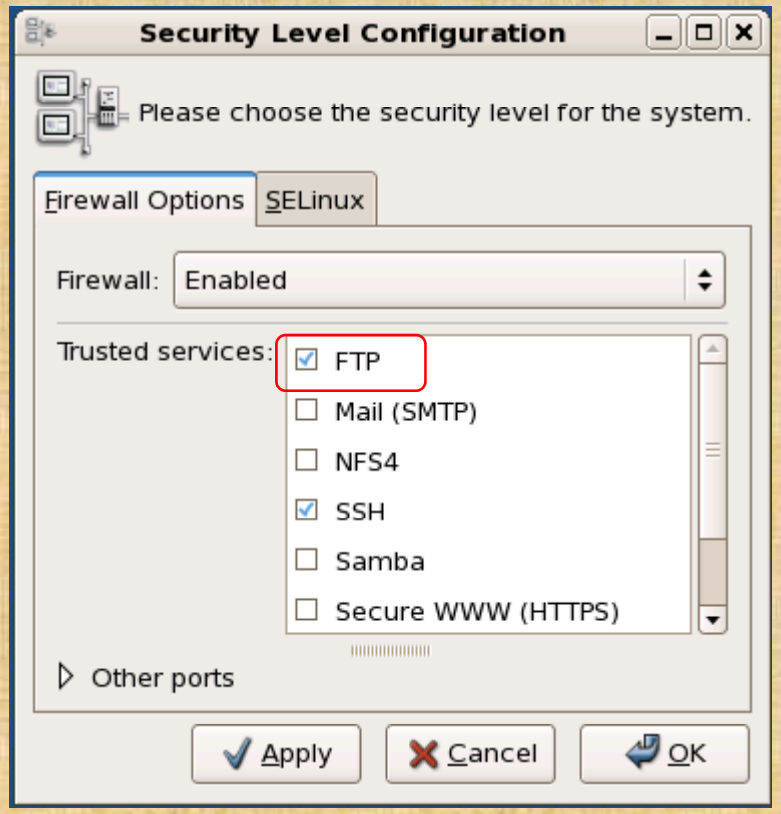


Step 4

Leave as Enforcing.

No changes are needed.

Activity



Arwen

On Arwen

1. Use **init 5** (to go to run level 5)

2. **System** > **Administration** > **Security Level and Firewall**
   - Check FTP as a trusted service
   - On SELinux tab, keep in Enforcing mode
   - Click OK

3. Check port 21 with **iptables –L –n | grep 21**
4. Check FTP connection tracking module with **lsmod | grep ftp**

Installing and Configuring vsftpd (Red Hat Family)

Step 5 *Start or restart service*

```
[root@bigserver ~]# service vsftpd start
Starting vsftpd for vsftpd:                                [  OK  ]
[root@bigserver ~]#
```

Step 6 *Automatically start at system boot*

```
[root@bigserver ~]# chkconfig vsftpd on
[root@bigserver ~]# chkconfig --list vsftpd
vsftpd          0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@bigserver ~]#
```

Activity

Arwen

Start the vsftpd service and configure it to run automatically at boot

service vsftpd start
service vsftpd status

chkconfig vsftpd on
chkconfig --list vsftpd

Installing and Configuring vsftpd

Step 7 *Verify service is running*

vsftpd processes

```
[root@arwen ~]# service vsftpd status
vsftpd (pid 7979 6475) is running...


[root@arwen ~]# ps -ef | grep vsftpd
root      6475     1  0 08:28 ?        00:00:00 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf
nobody    7975  6475  0 09:55 ?        00:00:00 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf
cis192    7979  7975  0 09:55 ?        00:00:00 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf
root      7995  7866  0 09:56 pts/3    00:00:00 grep vsftpd
[root@arwen ~]#
```

Individual vsftpd daemons are run for each session

Installing and Configuring vsftpd

netstat

```
[root@bigserver ~]# netstat -tln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 127.0.0.1:2208              0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:111                 0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:6000                0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:21                  0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:23                  0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:631               0.0.0.0:*                   LISTEN
tcp        0      0 0.0.0.0:792                 0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:25                0.0.0.0:*                   LISTEN
tcp        0      0 127.0.0.1:2207              0.0.0.0:*                   LISTEN
tcp        0      0 :::6000                     :::*                        LISTEN
tcp        0      0 :::22                       :::*                        LISTEN
[root@bigserver ~]#
```

Use netstat command to see what ports your system is listening for requests on

Installing and Configuring vsftpd

netstat

```
[root@bigserver ~]# netstat -tl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address               Foreign Address             State
tcp        0      0 r1.localdomain:2208        *:*                         LISTEN
tcp        0      0 *:sunrpc                   *:*                         LISTEN
tcp        0      0 *:x11                      *:*                         LISTEN
tcp        0      0 *:ftp                      *:*                         LISTEN
tcp        0      0 *:telnet                   *:*                         LISTEN
tcp        0      0 r1.localdomain:ipp         *:*                         LISTEN
tcp        0      0 *:792                      *:*                         LISTEN
tcp        0      0 r1.localdomain:smtp        *:*                         LISTEN
tcp        0      0 r1.localdomain:2207        *:*                         LISTEN
tcp        0      0 *:x11                      *:*                         LISTEN
tcp        0      0 *:ssh                      *:*                         LISTEN
[root@bigserver ~]#
```

Use netstat command to see what ports your system is listening for requests on

Installing and Configuring vsftpd

Try it! *Create sample files on Arwen*

```
[root@arwen ~]# cd /var/ftp/pub
[root@arwen pub]# echo Contents > file1
[root@arwen pub]# echo Contents > file2
[root@arwen pub]# chmod 644 *
[root@arwen pub]# ls -l
total 16
-rw-r--r-- 1 root root 9 Mar 17 09:09 file1
-rw-r--r-- 1 root root 9 Mar 17 09:09 file2
[root@arwen pub]#
```

Installing and Configuring vsftpd

Try it! *On Elrond, download the files using **lftp** client from Arwen*

```
[root@arwen ~]# lftp arwen
lftp arwen:~> ls
drwxr-xr-x    2 0        0            4096 Mar 17 15:22 pub
lftp arwen:/> cd pub
lftp arwen:/pub> ls
-rw-r--r--    1 0        0               9 Mar 17 15:22 file1
-rw-r--r--    1 0        0               9 Mar 17 15:22 file2
lftp arwen:/pub> mget file*
18 bytes transferred
Total 2 files transferred
lftp arwen:/pub> exit
[root@arwen ~]#
```

lftp is a ftp client that can run in the background, download multiple files at once and keep trying if the connection fails

Installing and Configuring vsftpd

Try it!

```
[root@elrond ~]# ftp arwen
Connected to arwen.localdomain.
220 Welcome to the SIMMS FTP service.
530 Please login with USER and PASS.
530 Please login with USER and PASS.
KERBEROS_V4 rejected as an authentication type
Name (arwen:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (127,0,0,1,239,167)
150 Here comes the directory listing.
drwxr-xr-x    2 0        0            4096 Mar 17 15:22 pub
226 Directory send OK.
ftp> cd pub
250 Directory successfully changed.
ftp> mget file*
mget file1? y
227 Entering Passive Mode (127,0,0,1,51,63)
150 Opening BINARY mode data connection for file1 (9 bytes).
226 File send OK.
9 bytes received in 3e-05 seconds (2.9e+02 Kbytes/s)
mget file2? y
227 Entering Passive Mode (127,0,0,1,107,2)
150 Opening BINARY mode data connection for file2 (9 bytes).
226 File send OK.
9 bytes received in 0.00017 seconds (52 Kbytes/s)
ftp> bye
221 Goodbye.
[root@elrond ~]#
```

*On Elrond, download the files using regular **ftp** client from Arwen*

Activity

On Arwen, create some sample files in the FTP root directory

Arwen

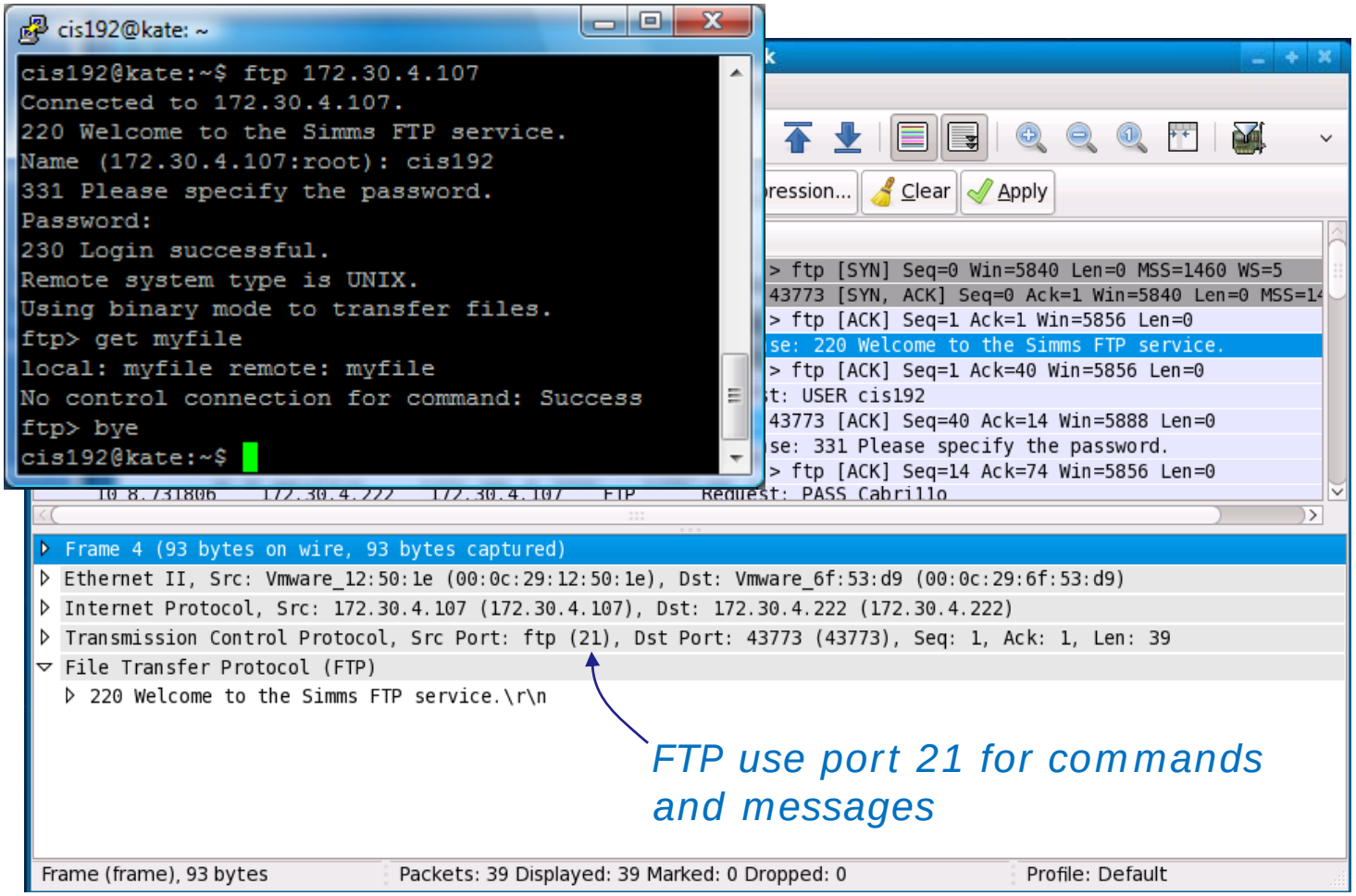
```
cd /var/ftp/pub
echo Contents > file1
echo Contents > file2
chmod 644 *
ls –l
```

On Elrond, download the sample files from Arwen

Elrond

```
lftp arwen
> ls
> cd pub
> ls
> mget file*
> exit
```

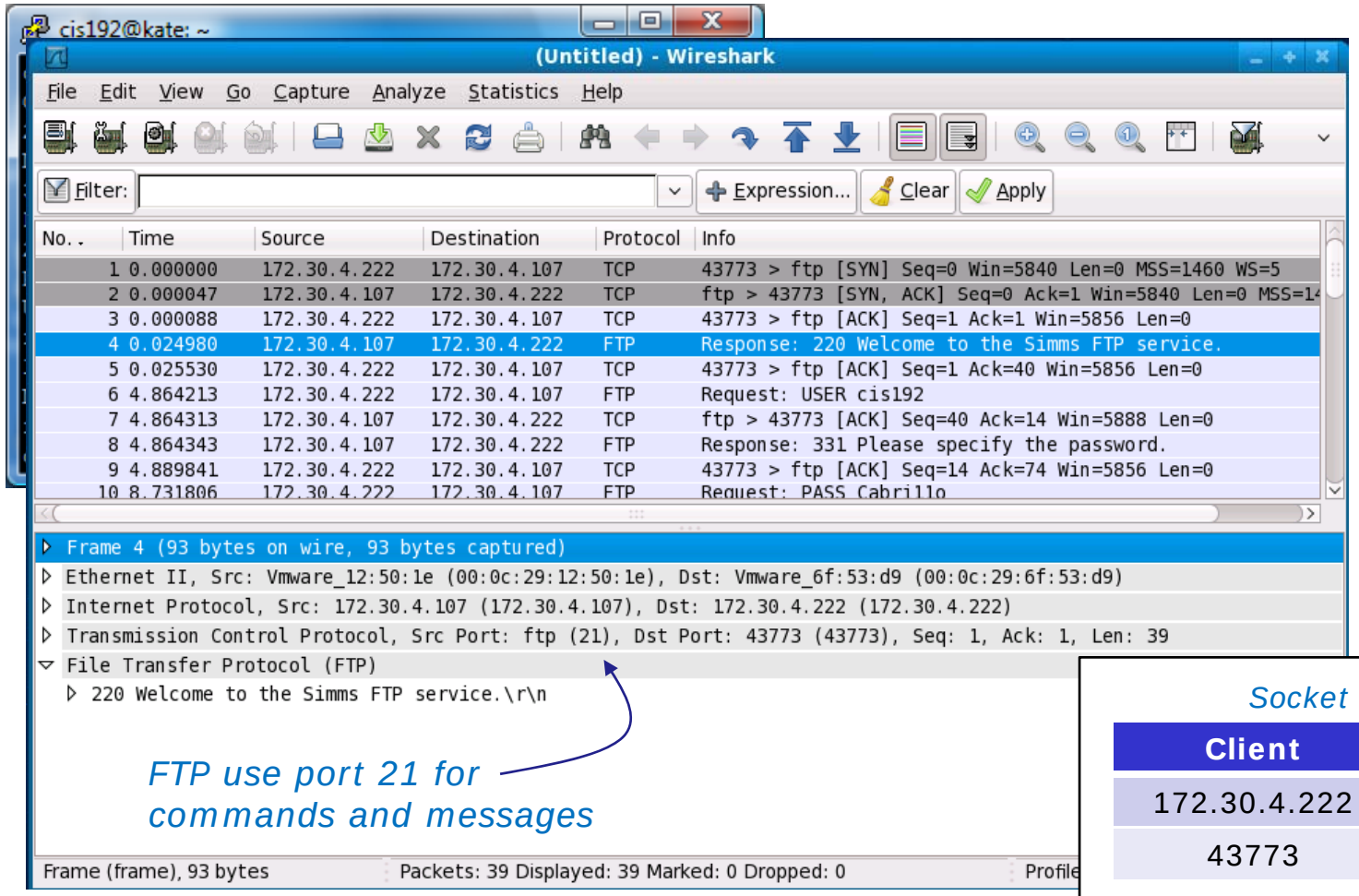
Installing and Configuring vsftpd



3-way handshake

Login is transmitted in clear text

Installing and Configuring vsftpd



FTP use port 21 for commands and messages

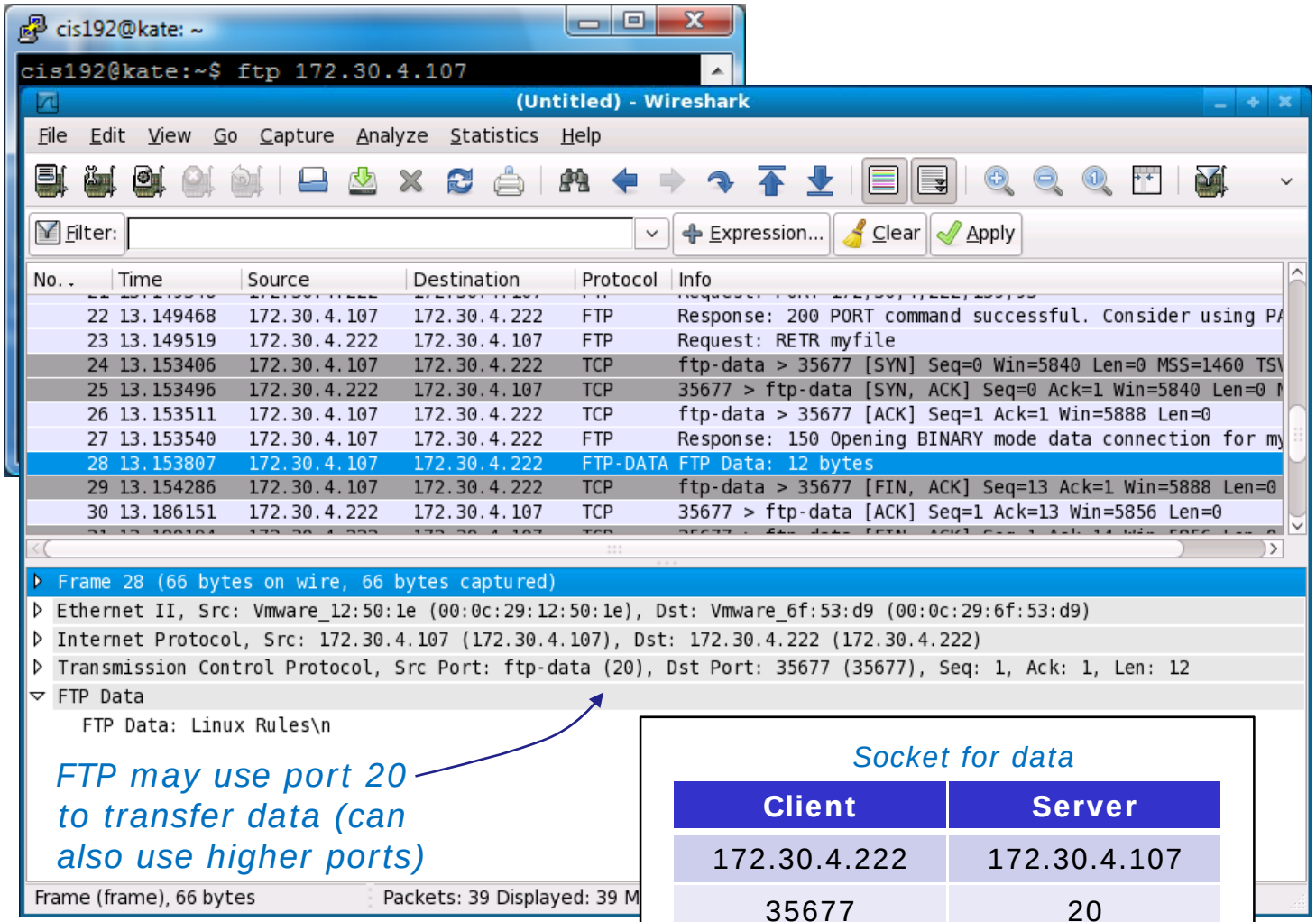
3-way handshake

Login is transmitted in clear text

Socket for commands

Client	Server
172.30.4.222	172.30.4.107
43773	21

Installing and Configuring vsftpd



FTP data (Layer 5) is encapsulated in a TCP segment

The TCP segment (layer 4) is encapsulated in an IP packet

The IP packet (layer 3) is encapsulated in Ethernet frame

The Ethernet frame (layer 2) is placed in a low level frame that travels via electrical signals on a physical cable (Layer 1)

Installing and Configuring vsftpd

Step 8 Troubleshooting

```
[root@elrond ~]# lftp arwen
lftp arwen:~> ls
`ls' at 0 [Delaying before reconnect: 27]
```

On the FTP server:

- *Check FTP service is running,*
- *Check TCP port 21 is open*
- *Check ip_conntrack_ftp kernel module is loaded*

Installing and Configuring vsftpd

Step 8 Troubleshooting

```
[root@elrond ~]# ftp arwen
ftp: connect: No route to host
ftp>
```

Open the firewall on the FTP sever to accept incoming FTP connections (TCP 21)

Use ***iptables -I RH-Firewall-1-INPUT 9 -m state --state NEW -m tcp -p tcp --dport 21 -j ACCEPT***

Installing and Configuring vsftpd

Step 8 Troubleshooting

```
[root@elrond ~]# ftp arwen
ftp: connect: Connection refused
ftp>
```

Make sure service is up and running on FTP server.
*Use **service vsftpd start***

Installing and Configuring vsftpd

Step 8 Troubleshooting

```
[root@elrond ~]# ftp arwen
Connected to arwen.
220 Welcome to the SIMMS FTP service.
530 Please login with USER and PASS.
530 Please login with USER and PASS.
KERBEROS_V4 rejected as an authentication type
Name (arwen:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (192,168,2,9,106,150)
ftp: connect: No route to host
ftp>
```

Make sure ip_conntrack_ftp kernel module has been loaded on FTP server. Use ***modprobe ip_conntrack_ftp***

Installing and Configuring vsftpd

Step 9 Monitor log files

```
[root@arwen ~]# tail -f /var/log/xferlog
Wed Mar 17 15:50:41 2010 1 127.0.0.1 9 /pub/file1 b _ o a lftp@ ftp 0 * c
Wed Mar 17 15:50:41 2010 1 127.0.0.1 9 /pub/file2 b _ o a lftp@ ftp 0 * c
Wed Mar 17 16:03:00 2010 1 127.0.0.1 9 /pub/file1 b _ o a ? ftp 0 * c
Wed Mar 17 16:03:01 2010 1 127.0.0.1 9 /pub/file2 b _ o a ? ftp 0 * c
Wed Mar 17 16:35:06 2010 1 192.168.2.1 0 /pub/f* b _ o a lftp@ ftp 0 * i
Wed Mar 17 16:35:17 2010 1 192.168.2.1 0 /pub/file* b _ o a lftp@ ftp 0 * i
Wed Mar 17 16:35:21 2010 1 192.168.2.1 9 /pub/file1 b _ o a lftp@ ftp 0 * c
Wed Mar 17 16:35:21 2010 1 192.168.2.1 9 /pub/file2 b _ o a lftp@ ftp 0 * c
Wed Mar 17 16:39:27 2010 1 192.168.2.1 9 /pub/file1 b _ o a ? ftp 0 * c
Wed Mar 17 16:39:28 2010 1 192.168.2.1 9 /pub/file2 b _ o a ? ftp 0 * c
```

```
[root@arwen ~]# cat /var/log/secure | grep -i vsftpd
Mar 17 07:47:27 arwen vsftpd: pam_unix(vsftpd:auth): authentication failure;
logname= uid=0 euid=0 tty=ftp ruser=cis192 rhost=elrond  user=cis192
Mar 17 08:02:56 arwen vsftpd: pam_unix(vsftpd:auth): authentication failure;
logname= uid=0 euid=0 tty=ftp ruser=cis192 rhost=elrond  user=cis192
[root@arwen ~]#
```

Installing and Configuring vsftpd

Does vsftpd use TCP Wrappers?

```
[root@bigserver ~]# type vsftpd
vsftpd is /usr/sbin/vsftpd
[root@bigserver ~]# ldd /usr/sbin/vsftpd
        linux-gate.so.1 =>  (0x0074c000)
        libssl.so.6 => /lib/libssl.so.6 (0x0012a000)
        libwrap.so.0 => /usr/lib/libwrap.so.0 (0x005cb000)
        libnsl.so.1 => /lib/libnsl.so.1 (0x00913000)
        libpam.so.0 => /lib/libpam.so.0 (0x00b11000)
        libcap.so.1 => /lib/libcap.so.1 (0x0084a000)
        libdl.so.2 => /lib/libdl.so.2 (0x00110000)
        libc.so.6 => /lib/libc.so.6 (0x0016f000)
        libcrypto.so.6 => /lib/libcrypto.so.6 (0x002b2000)
        libgssapi_krb5.so.2 => /usr/lib/libgssapi_krb5.so.2 (0x00bb4000)
        libkrb5.so.3 => /usr/lib/libkrb5.so.3 (0x003e5000)
        libcom_err.so.2 => /lib/libcom_err.so.2 (0x0092c000)
        libk5crypto.so.3 => /usr/lib/libk5crypto.so.3 (0x0054c000)
        libresolv.so.2 => /lib/libresolv.so.2 (0x00114000)
        libz.so.1 => /usr/lib/libz.so.1 (0x00478000)
        libaudit.so.0 => /lib/libaudit.so.0 (0x004c5000)
        /lib/ld-linux.so.2 (0x0085a000)
        libkrb5support.so.0 => /usr/lib/libkrb5support.so.0 (0x00fb5000)
        libkeyutils.so.1 => /lib/libkeyutils.so.1 (0x00961000)
        libselinux.so.1 => /lib/libselinux.so.1 (0x0048b000)
        libsepol.so.1 => /lib/libsepol.so.1 (0x004da000)
[root@bigserver ~]#
```

yes it does

Installing and Configuring vsftpd

Step 10 *Configure additional security with TCP wrappers*

TCP Wrappers and vsftpd

vsftpd is compiled with TCP wrappers

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

Installing and Configuring vsftpd

TCP Wrappers and vsftpd 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
```

For vsftpd, only Frodo, Arwen and Sauron hosts are allowed

Nosmo at 172.30.1.1 is NOT included

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

Everyone else is denied (this includes Nosmo)

Installing and Configuring vsftpd

TCP Wrappers and vsftpd 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

Access permitted

```
root@sauron:~# ftp arwen
Connected to arwen.
220 Welcome to the Cabrillo Super FTP service.
Name (arwen:cis192): cis192
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> bye
221 Goodbye.
root@sauron:~#
```

Nosmo

Access denied

```
[root@nosmo root]# ftp 192.168.2.9
Connected to 192.168.2.9 (192.168.2.9).
421 Service not available.
ftp>
```

super daemons

Super Daemons

The Super Daemons

There are three primary super-daemons controlling server services. Super daemons spawn other daemons to handle specific client requests.

- **inetd** - From UNIX days, this was the primary daemon for handling tcp application services. It is being replaced by xinetd.
- **portmap** - portmapper operates with Remote Procedure Call (RCP) applications.
- **xinetd** - Extended Internet Services Daemon: used by modern distributions of Linux.

Super Daemons

xinetd Daemon

Advantages

1. Provides access control for TCP, UDP, and RPC services
2. Access limitations based on time
3. Extensive logging capabilities
4. Provides numerous mechanisms to prevent denial of service attacks
   - can limit number of daemons that can run
   - can limit number or processes forked by xinetd
   - can limit log file sizes
5. Supports TCP_Wrappers through libwrap
6. Services may be bound to specific interfaces
7. Services may be forwarded (proxied) to another system
8. Supports ipv6

Super Daemons

xinetd

Syntax:

```
service service_name
{
        attribute operator value value ...
}
```

Example:

```
[root@arwen ~]# cat /etc/xinetd.d/telnet
# default: on
# description: The telnet server serves telnet sessions; it uses \
#       unencrypted username/password pairs for authentication.
service telnet
{
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/in.telnetd
        log_on_failure  += USERID
        disable         = no
}
```

Application Layer

xinetd required attributes

- socket_type — ***stream*** *for TCP,* ***dgram*** *for UDP*
- wait — *Use* ***no*** *for single threaded devices where you must wait until first daemon finishes before starting another, otherwise* ***yes*** *for multiple concurrent daemons running*
- user — *user the daemon runs as*
- server — *path to daemon*
- port
- protocol

  } *if not specified, uses default for service*
- rpc_version, rpc_number

Application Layer

Some aditional xinetd attributes

- Access attributes
  - only_from *hosts to allow*
  - no_access *hosts to deny*
- The bind attribute *(IP address) to limit service to specific interface on services with multiple interfaces*
- The redirect attribute *(IP address and port) to redirect to another server*

Super Daemons

Some xinetd command line options

1. -d *Debug mode*
2. -syslog *specify logging facility*
3. -loop rate *specify numbers of daemons forked per second*
4. -reuse *Reusable TCP socket for allow multiple concurrent daemons*
5. -limit *limit number of processes started by xinetd*

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 ~]# aptitude update
[root@sauron ~]# aptitude 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

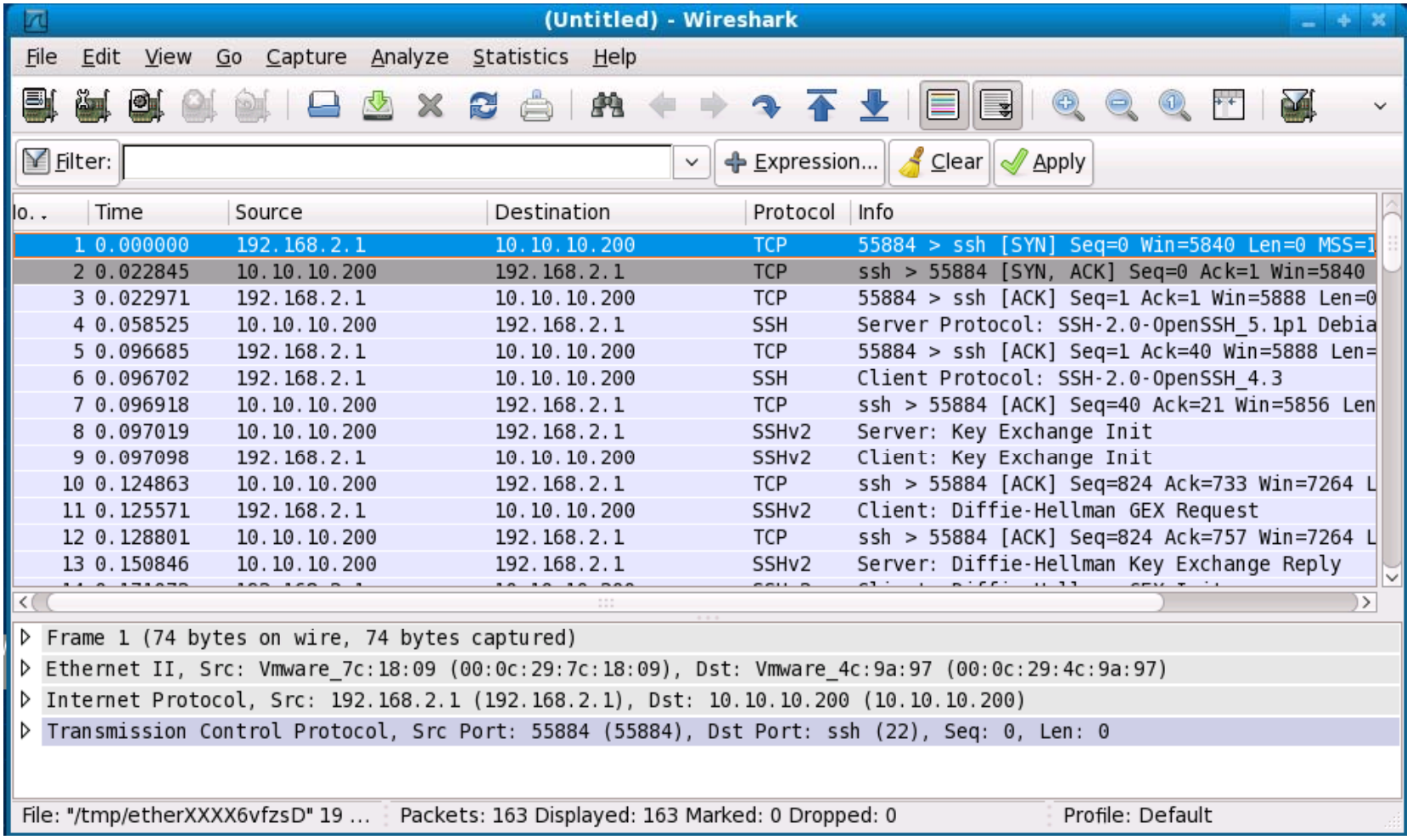
```
root@sauron:~# ps -ef | grep ssh
root      7601     1  0 13:59 ?        00:00:00 /usr/sbin/sshd
root      7607  7601  1 14:11 ?        00:00:00 sshd: root@pts/2
root      7632  7601  1 14:11 ?        00:00:00 sshd: root@pts/3
root      7658  7280  0 14:12 pts/1    00:00:00 grep ssh


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 ~]#
```

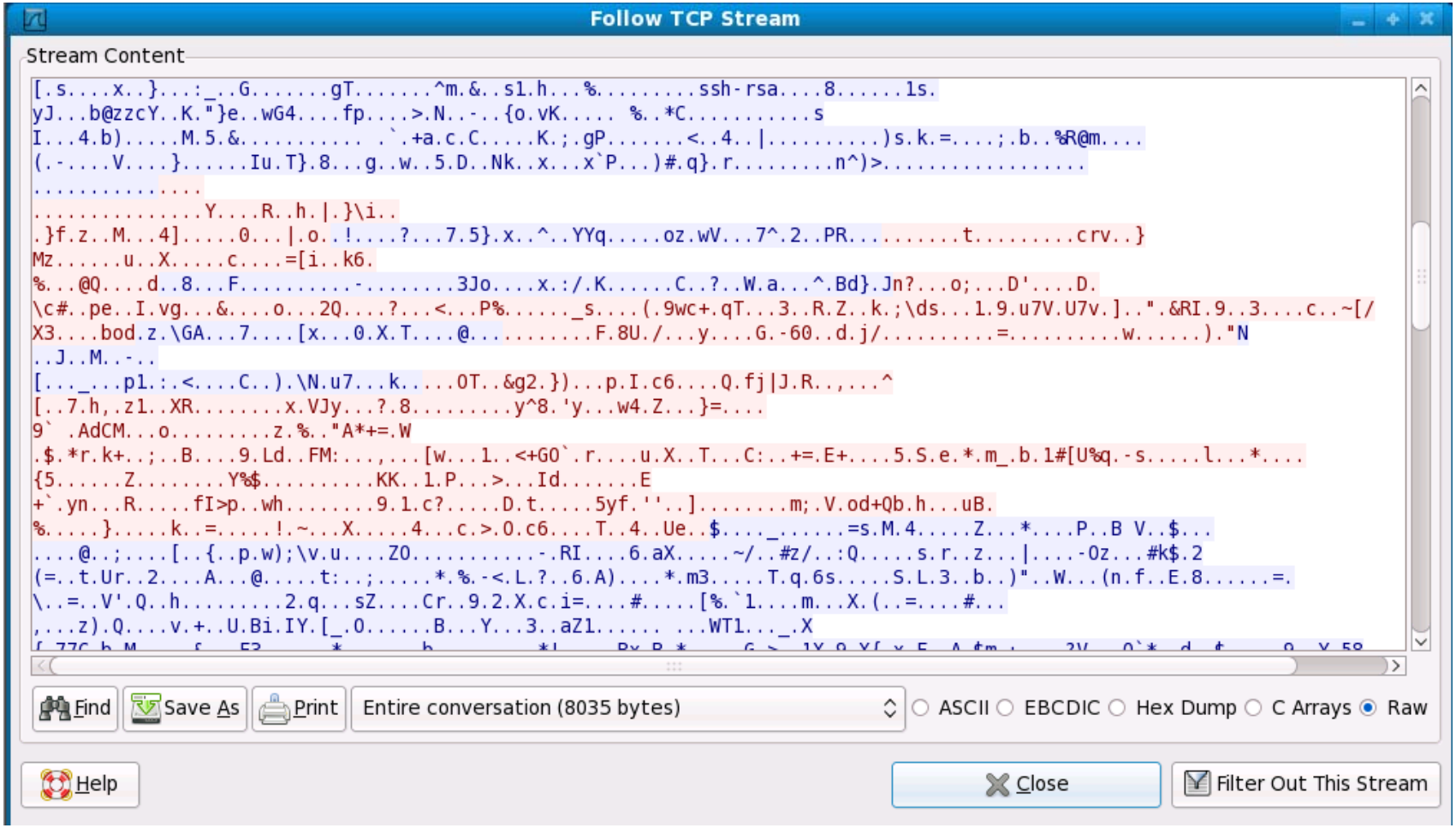
sshd



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

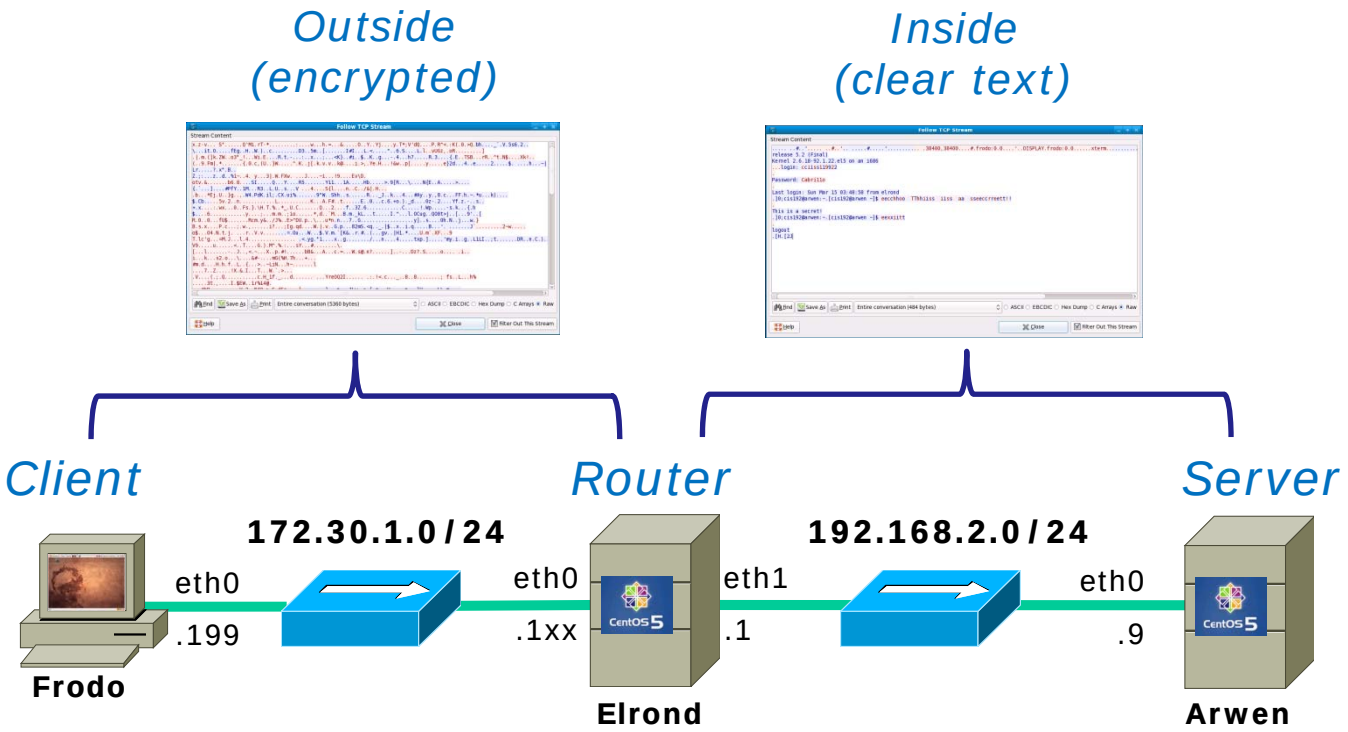
Nosmo

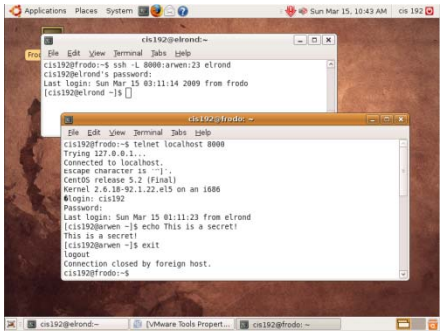
```
[root@nosmo root]# ssh 192.168.2.9
ssh_exchange_identification: Connection closed by remote host
[root@nosmo root]#
```

Access denied

SSH Port Forwarding

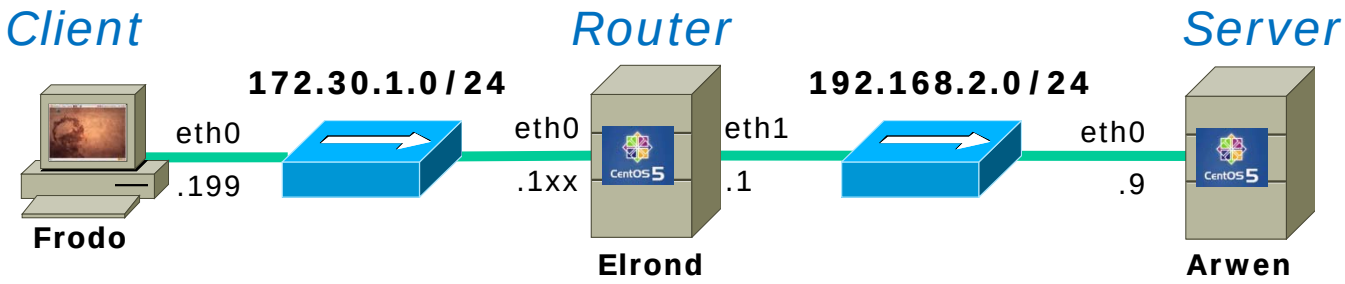
SSH Port Forwarding

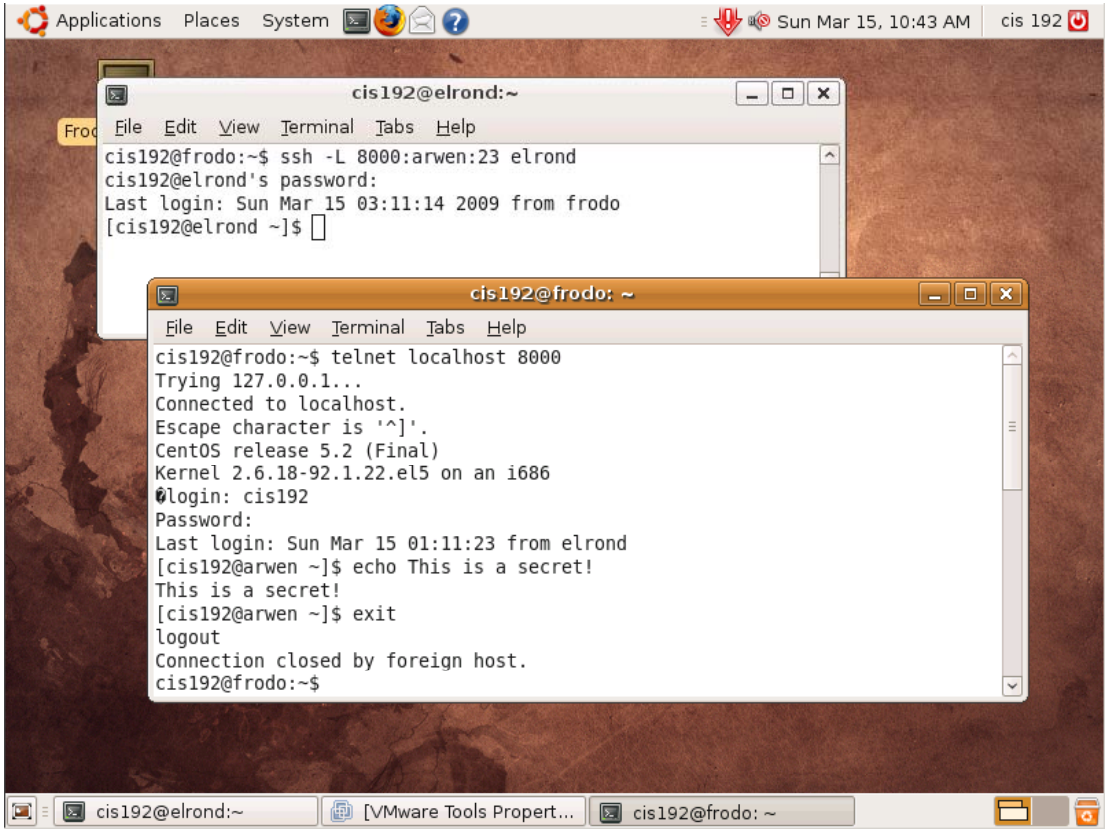




In this example we will tunnel a telnet session through an encrypted SSH connection.

SSH Port Forwarding





Requires one Frodo terminal to setup SSH port forwarding

And another Frodo terminal to make the Telnet connection

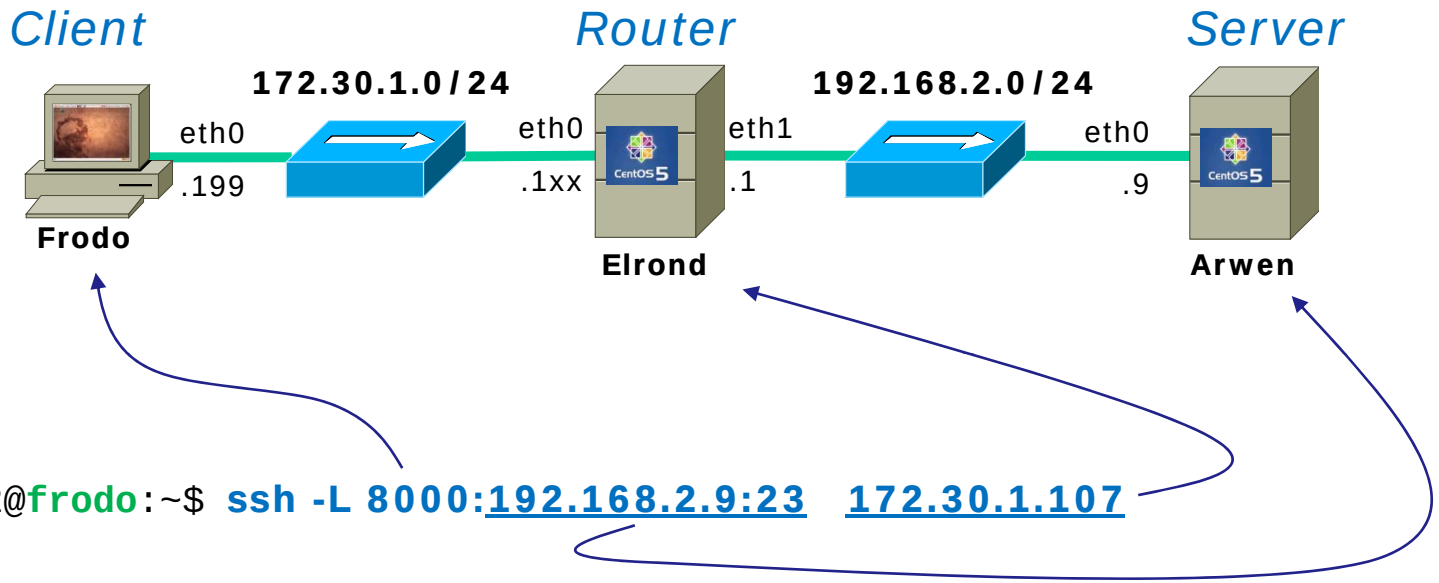
SSH Port Forwarding

-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.

The –L option on the SSH command

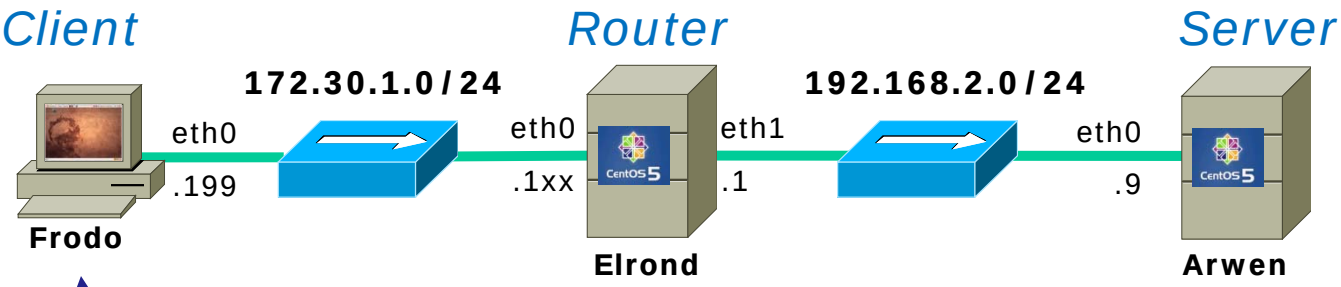
SSH Port Forwarding



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.

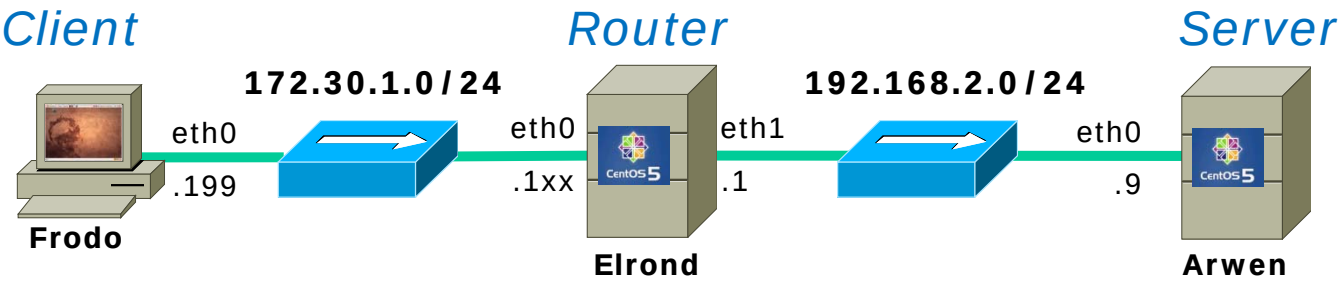
SSH Port Forwarding




```
cis192@frodo:~$ ssh -L 8000:arwen:23 elrond
```

Same as before, just using names instead of IP addresses which have been configured in /etc/hosts

SSH Port Forwarding



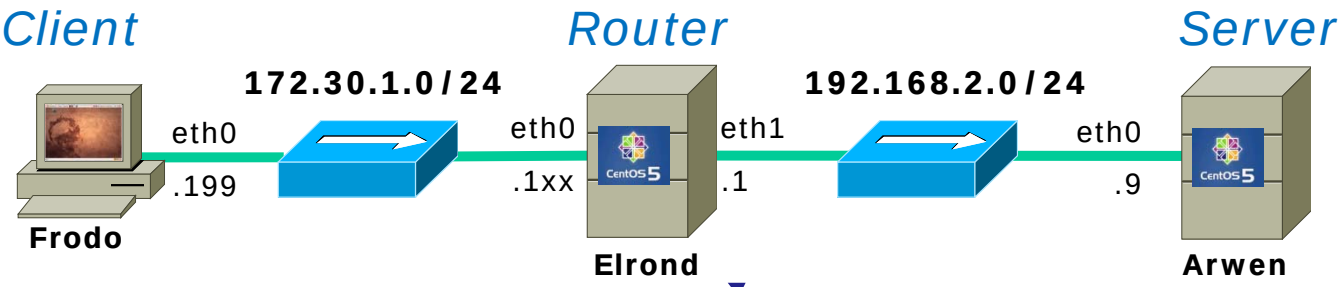
Use –L option to enable port forwarding

```
cis192@frodo:~$ ssh -L 8000:192.168.2.9:23 172.30.1.107
The authenticity of host '172.30.1.107 (172.30.1.107)' can't be established.
RSA key fingerprint is 54:56:16:38:a3:ad:1d:d2:62:5b:26:de:60:06:98:f7.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.30.1.107' (RSA) to the list of known hosts.
cis192@172.30.1.107's password:
Last login: Fri Mar 13 04:50:57 2009 from 172.30.1.199
[cis192@elrond ~]$
[cis192@elrond ~]$ exit
logout

Connection to 172.30.1.107 closed.

cis192@frodo:~$
```

Port forwarding enabled

Port forwarding disabled after exit

SSH Port Forwarding



Use –L option to enable port forwarding

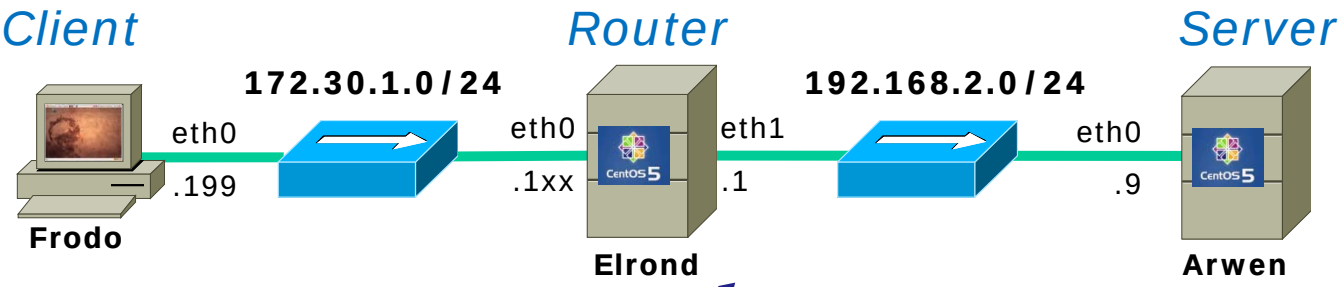
```
cis192@frodo:~$ ssh -L 8000:192.168.2.9:23 elrond
The authenticity of host 'elrond (172.30.1.107)' can't be established.
RSA key fingerprint is 54:56:16:38:a3:ad:1d:d2:62:5b:26:de:60:06:98:f7.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'elrond' (RSA) to the list of known hosts.
cis192@elrond's password:
Last login: Sun Mar 15 03:09:45 2009 from frodo
[cis192@elrond ~]$ exit
logout

Connection to elrond closed.

cis192@frodo:~$
```

Port forwarding enabled

Port forwarding disabled

This is the same as the last slide except using a host name for elrond

SSH Port Forwarding



Use –L option to enable port forwarding

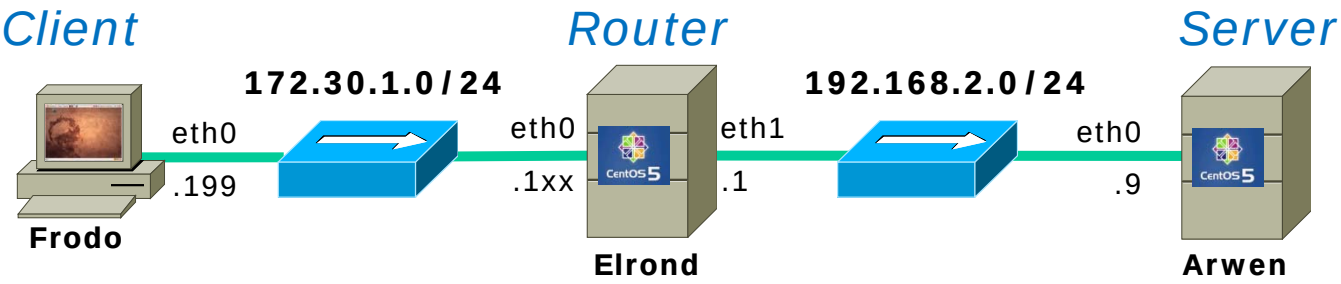
```
cis192@frodo:~$ ssh -L 8000:arwen:23 elrond
cis192@elrond's password:
Last login: Sun Mar 15 03:11:14 2009 from frodo
[cis192@elrond ~]$
[cis192@elrond ~]$
[cis192@elrond ~]$ exit
logout

Connection to elrond closed.

cis192@frodo:~$
```

Port forwarding enabled

Port forwarding disabled

This is the same as the last slide except using a host name for elrond and arwen

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

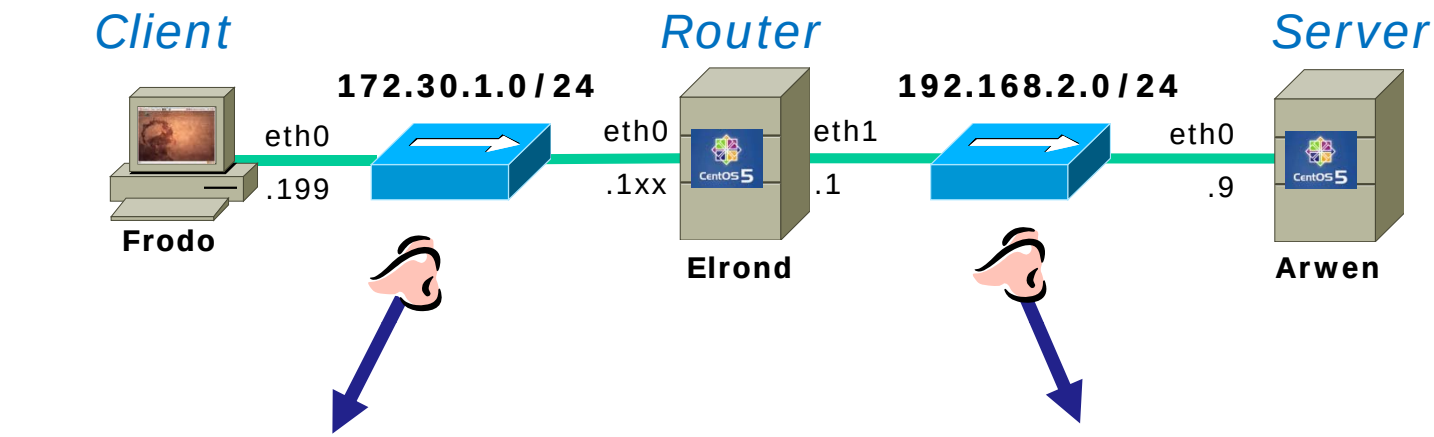
Enable port forwarding in first terminal

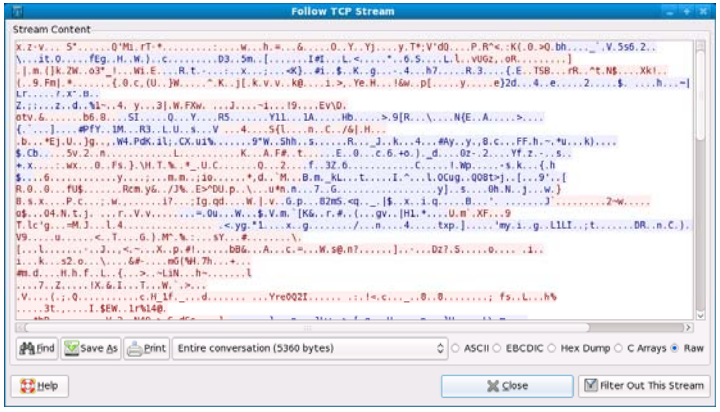
```
cis192@elrond:~
File  Edit  View  Terminal  Tabs  Help
cis192@frodo:~$ ssh -L 8000:arwen:23 elrond
cis192@elrond's password:
Last login: Sun Mar 15 03:11:14 2009 from frodo
[cis192@elrond ~]$
```

Use port forwarding in second terminal

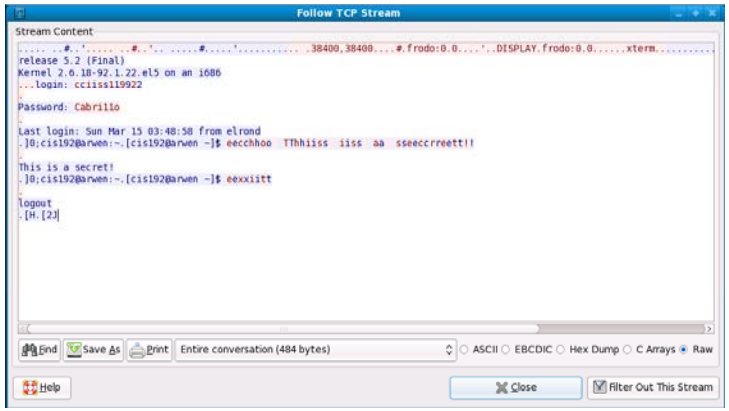
```
cis192@frodo: ~
File  Edit  View  Terminal  Tabs  Help
cis192@frodo:~$ telnet localhost 8000
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
CentOS release 5.2 (Final)
Kernel 2.6.18-92.1.22.el5 on an i686
login: cis192
Password:
Last login: Sun Mar 15 03:48:58 from elrond
[cis192@arwen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


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

SSH Port Forwarding

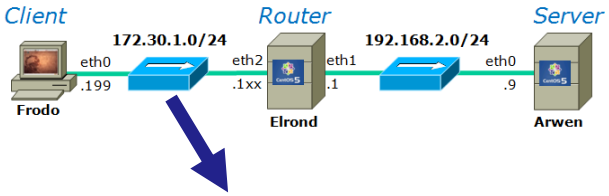




This portion is encrypted



This portion is in clear text



SSH Port Forwarding

Encrypted portion of the connection

(Untitled) - Wireshark

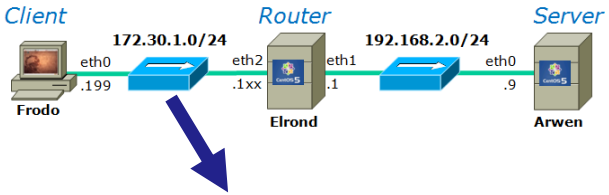
File Edit View Go Capture Analyze Statistics Help

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

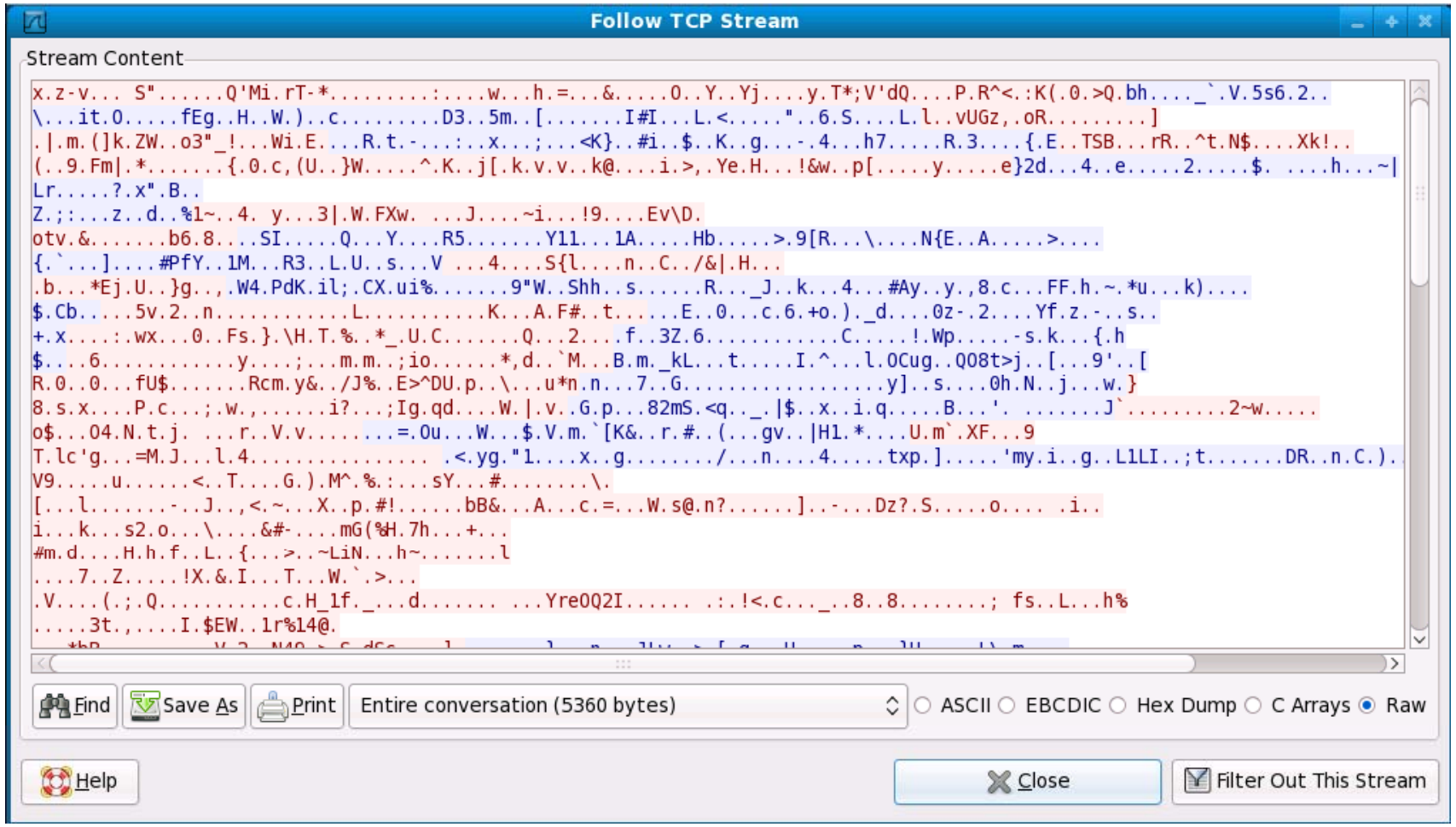
Frame 10 (118 bytes on wire, 118 bytes captured)
Ethernet II, Src: Vmware_4e:21:af (00:0c:29:4e:21:af), Dst: Vmware_6f:53:d9 (00:0c:29:6f:53:d9)
Internet Protocol, Src: 172.30.4.107 (172.30.4.107), Dst: 172.30.4.199 (172.30.4.199)
Transmission Control Protocol, Src Port: ssh (22), Dst Port: 44022 (44022),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

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

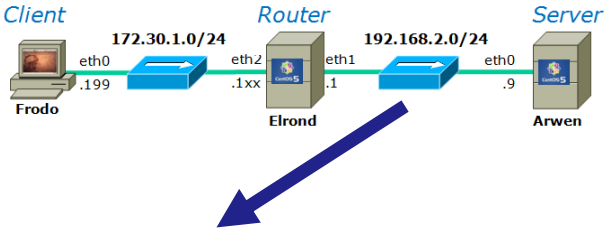


SSH Port Forwarding

Encrypted portion of the connection



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

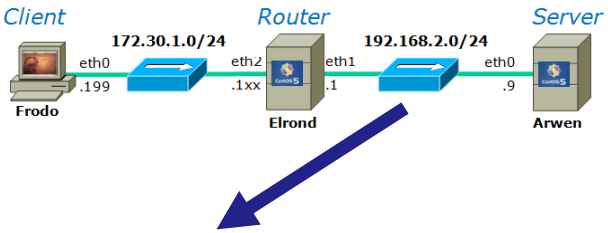
Command: Will Echo

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

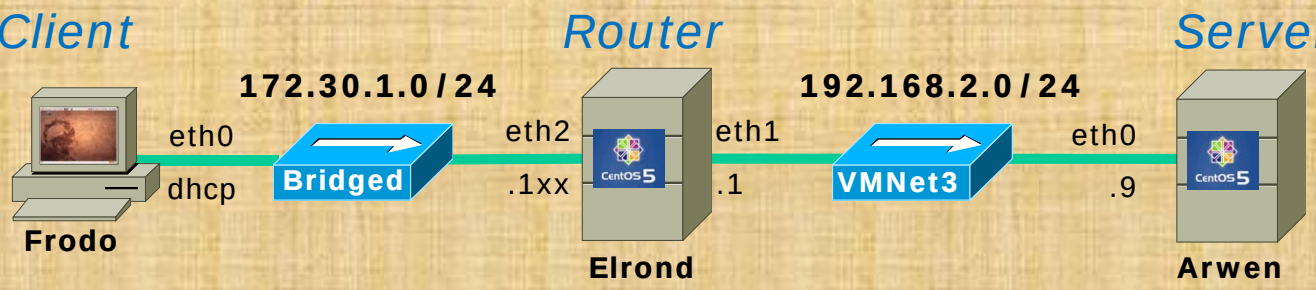
Stream Content

```
..... ..#..'..... ..#..'.. .....#.....'.......... .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: Cabri11o
.
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

Exercise



On Frodo:

1. On first terminal
   - Add **172.30.1.1xx elrond** and **192.168.2.9 arwen** to **/etc/hosts**
   - Enable port forwarding with **ssh –L 8000:arwen:23 elrond**

2. On second terminal
   - Tunnel to Arwen with **telnet localhost 8000**

3. On first terminal, exit to discontinue port forwarding

TCP Wrappers

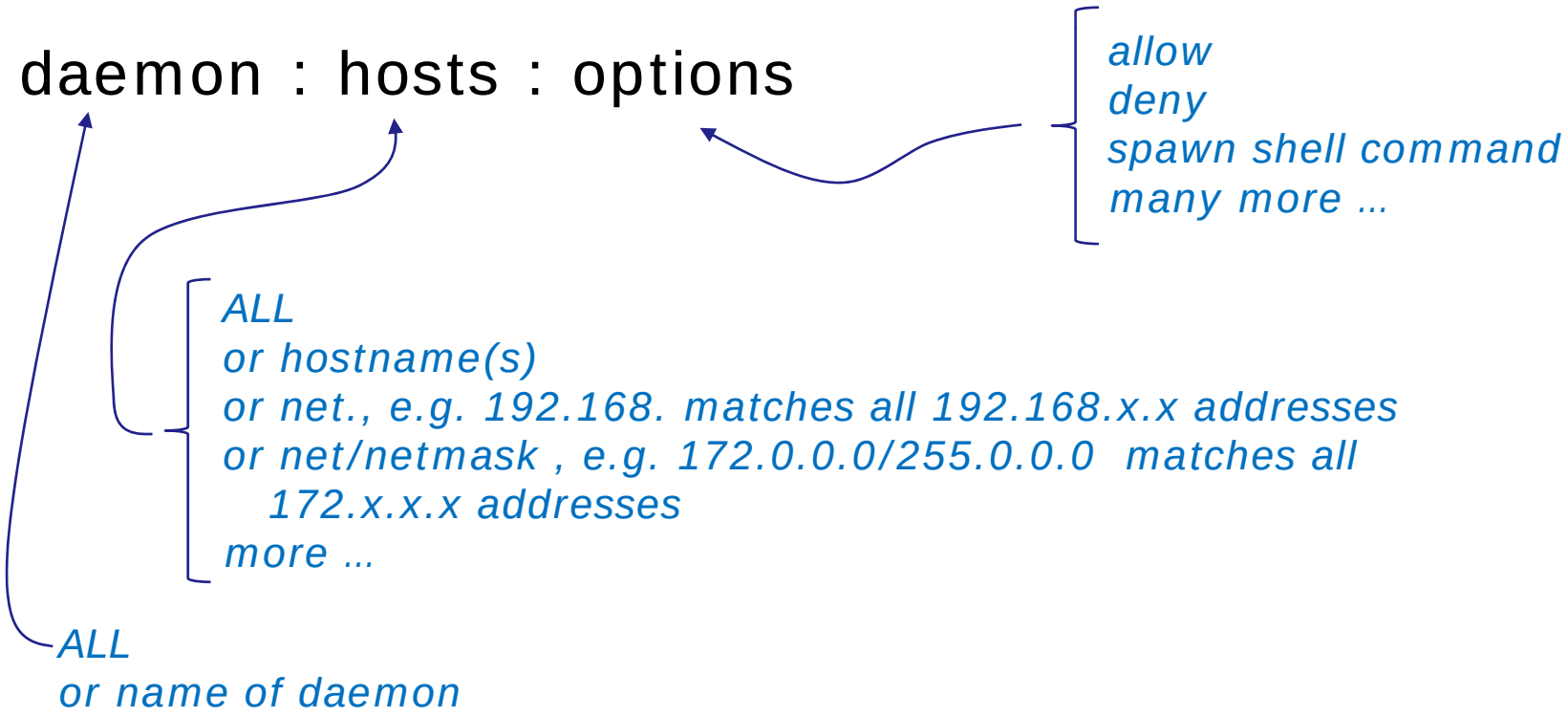
TCP Wrappers

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

TCP Wrappers

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



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

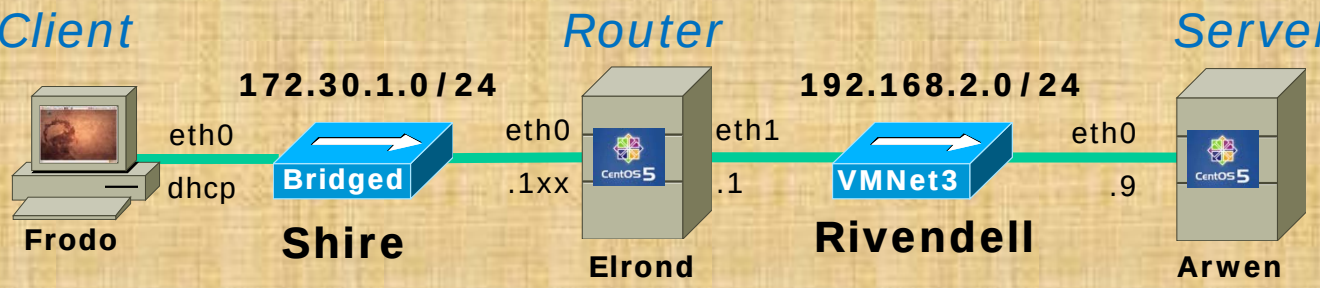
TCP Wrappers

Access controls

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

```
[root@arwen ~]# type tcpd
tcpd is /usr/sbin/tcpd
[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 ~]#
```

Exercise



/etc/hosts.allow
in.telnetd: 172.30.

/etc/hosts.deny
ALL: ALL

- On Frodo, add static route with:

  route add –net 192.168.2.0/24 gw 172.30.1.1xx

- Configure Telent on Arwen to allow Shire hosts and block Rivendell hosts
- Configure Arwen to allow telnet connections from Frodo but not Arwen
- Test Telnet connections from Frodo and Elrond
- When finished, on Arwen, comment out **/etc/hosts.deny** and **/etc/hosts.allow**
- When finished, on Frodo, remove static route

  route del –net 192.168.2.0/24 gw 172.30.1.1xx

Netfilter

Using iptables for firewalls and NAT

Examples

Firewalls and NAT

- Lets first look at some actual firewall and NAT configuration in all their complexity
- Then we will start breaking it down step-by-step

Default Red Hat Firewall

Current settings

```
[root@elrond ~]# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0           icmp type 255
ACCEPT     esp  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     ah   --  0.0.0.0/0            0.0.0.0/0
ACCEPT     udp  --  0.0.0.0/0            224.0.0.251         udp dpt:5353
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:631
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:631
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-
prohibited
[root@elrond ~]#
```

Default Red Hat Firewall

```
[root@elrond ~]# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0           icmp type 255
ACCEPT     esp  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     ah   --  0.0.0.0/0            0.0.0.0/0
ACCEPT     udp  --  0.0.0.0/0            224.0.0.251         udp dpt:5353
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:631
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:631
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-
prohibited
[root@elrond ~]#
```

The three standard filter chains and one custom chain

Default Red Hat Firewall

```
[root@elrond ~]# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0           icmp type 255
ACCEPT     esp  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     ah   --  0.0.0.0/0            0.0.0.0/0
ACCEPT     udp  --  0.0.0.0/0            224.0.0.251         udp dpt:5353
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:631
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:631
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-
prohibited
[root@elrond ~]#
```

The policy on the three filter chains is ACCEPT.

The policy is the final rule in the chain and is used when no other rules in the chain apply.

Default Red Hat Firewall

```
[root@elrond ~]# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0           icmp type 255
ACCEPT     esp  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     ah   --  0.0.0.0/0            0.0.0.0/0
ACCEPT     udp  --  0.0.0.0/0            224.0.0.251         udp dpt:5353
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:631
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:631
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-
prohibited
[root@elrond ~]#
```

The INPUT and FORWARD filter chains have no rules of their own, they will use the rules in same custom chain named RH-Firewall-1-INPUT

Default Red Hat Firewall

```
[root@elrond ~]# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0           icmp type 255
ACCEPT     esp  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     ah   --  0.0.0.0/0            0.0.0.0/0
ACCEPT     udp  --  0.0.0.0/0            224.0.0.251         udp dpt:5353
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:631
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:631
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-
prohibited
[root@elrond ~]#
```

Accept all traffic that arrives on the loopback interface.

Its not obvious from this output but the details can be seen in /etc/sysconfig/iptables

Default Red Hat Firewall

```
[root@elrond ~]# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0           icmp type 255
ACCEPT     esp  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     ah   --  0.0.0.0/0            0.0.0.0/0
ACCEPT     udp  --  0.0.0.0/0            224.0.0.251         udp dpt:5353
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:631
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:631
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-
prohibited
[root@elrond ~]#
```

All ICMP protocol traffic (of any type) is allowed.

Default Red Hat Firewall

```
[root@elrond ~]# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0           icmp type 255
ACCEPT     esp  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     ah   --  0.0.0.0/0            0.0.0.0/0
ACCEPT     udp  --  0.0.0.0/0            224.0.0.251         udp dpt:5353
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:631
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:631
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-
prohibited
[root@elrond ~]#
```

All ESP and AH protocol traffic is allowed.

ESP (Encapsulating Security Payload) and AH (Authentication Header) are used for IPsec.

Default Red Hat Firewall

```
[root@elrond ~]# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0           icmp type 255
ACCEPT     esp  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     ah   --  0.0.0.0/0            0.0.0.0/0
ACCEPT     udp  --  0.0.0.0/0            224.0.0.251         udp dpt:5353
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:631
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:631
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-
prohibited
[root@elrond ~]#
```

All multicast DNS traffic to port 5353 is allowed.

This is used with zeroconf (Zero configuration networking) to locate DNS services on small LANs .

Default Red Hat Firewall

```
[root@elrond ~]# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0           icmp type 255
ACCEPT     esp  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     ah   --  0.0.0.0/0            0.0.0.0/0
ACCEPT     udp  --  0.0.0.0/0            224.0.0.251         udp dpt:5353
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:631
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:631
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-
prohibited
[root@elrond ~]#
```

All UDP and TCP protocol traffic to port 631 is allowed.

This allows CUPS to listen for IPP (Internet Printing Protocol) requests.

Default Red Hat Firewall

```
[root@elrond ~]# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0           icmp type 255
ACCEPT     esp  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     ah   --  0.0.0.0/0            0.0.0.0/0
ACCEPT     udp  --  0.0.0.0/0            224.0.0.251         udp dpt:5353
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:631
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:631
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-
prohibited
[root@elrond ~]#
```

Any traffic whose connection was locally originated or related to that connection is allowed

Default Red Hat Firewall

```
[root@elrond ~]# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0           icmp type 255
ACCEPT     esp  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     ah   --  0.0.0.0/0            0.0.0.0/0
ACCEPT     udp  --  0.0.0.0/0            224.0.0.251         udp dpt:5353
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:631
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:631
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-
prohibited
[root@elrond ~]#
```

Any new incoming connections to port 22 (ssh) are allowed

Default Red Hat Firewall

```
[root@elrond ~]# iptables -L -n
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0           icmp type 255
ACCEPT     esp  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     ah   --  0.0.0.0/0            0.0.0.0/0
ACCEPT     udp  --  0.0.0.0/0            224.0.0.251         udp dpt:5353
ACCEPT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpt:631
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:631
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:22
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-
prohibited
[root@elrond ~]#
```

If any of the previous rules did not apply, then send an error back using ICMP

Default Red Hat Firewall

```
[root@elrond ~]# cat /etc/sysconfig/iptables
# Generated by iptables-save v1.3.5 on Wed Mar 17 12:04:26 2010
*nat
:PREROUTING ACCEPT [1:94]
:POSTROUTING ACCEPT [6:994]
:OUTPUT ACCEPT [6:994]
COMMIT
# Completed on Wed Mar 17 12:04:26 2010
# Generated by iptables-save v1.3.5 on Wed Mar 17 12:04:26 2010
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [34:7149]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -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 -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Wed Mar 17 12:04:26 2010
[root@elrond ~]#
```

Permanent settings to be used at next system boot or when restarting the iptables service

Shows the actual iptables commands used to create the firewall

Default Red Hat Firewall

Backup the permanent settings

```
[root@elrond ~]# cp /etc/sysconfig/iptables /etc/sysconfig/iptables.bak
```

Save the current firewall and NAT settings for use at next reboot

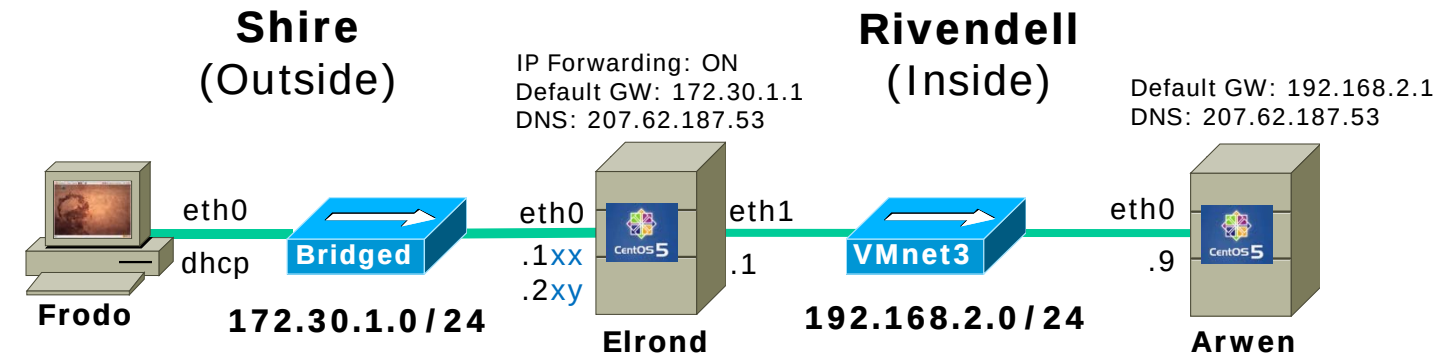
```
[root@elrond ~]# iptables-save > /etc/sysconfig/iptables
```

Start using the rules saved in /etc/sysconfig/iptables

```
[root@elrond ~]# service iptables restart
Flushing firewall rules:                                   [  OK  ]
Setting chains to policy ACCEPT: filter nat                [  OK  ]
Unloading iptables modules:                                [  OK  ]
Applying iptables firewall rules:                          [  OK  ]
Loading additional iptables modules: ip_conntrack_netbios_n[  OK  ]
[root@elrond ~]#
```

Just like IP addresses we can set firewall and NAT rules temporarily or permanently.

Firewall and NAT settings for Lab 5



Firewall and NAT settings for Lab 5

Elrond

```
[root@elrond ~]# iptables -L -n
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  192.168.2.0/24       192.168.2.1         state NEW
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
LOG        all  --  0.0.0.0/0            0.0.0.0/0           LOG flags 0 level 6 prefix
`iptables INPUT: '

Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  192.168.2.0/24       0.0.0.0/0           state NEW
ACCEPT     tcp  --  0.0.0.0/0            192.168.2.9         state
NEW,RELATED,ESTABLISHED tcp dpt:23
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
LOG        all  --  0.0.0.0/0            0.0.0.0/0           LOG flags 0 level 6 prefix
`iptables FORWARD: '

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state
NEW,RELATED,ESTABLISHED
```

Firewall and NAT settings for Lab 5

Elrond

```
[root@elrond ~]# iptables -L -n -t nat
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
DNAT       all  --  0.0.0.0/0            172.30.1.200        to:192.168.2.9

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
SNAT       all  --  192.168.2.9          0.0.0.0/0           to:172.30.1.200
SNAT       all  --  192.168.2.0/24       0.0.0.0/0           to:172.30.1.121

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@elrond ~]#
```

Note, using classroom addresses for this example

Firewall and NAT settings for Lab 5

Elrond

```
[root@elrond sysconfig]# cat iptables.lab5
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*nat
:PREROUTING ACCEPT [376:36875]
:POSTROUTING ACCEPT [74:4747]
:OUTPUT ACCEPT [60:3780]
-A PREROUTING -d 172.30.4.122 -i eth0 -j DNAT --to-destination 192.168.2.9
-A POSTROUTING -s 192.168.2.9 -o eth0 -j SNAT --to-source 172.30.4.122
-A POSTROUTING -s 192.168.2.0/255.255.255.0 -o eth0 -j SNAT --to-source 172.30.4.121
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*filter
:INPUT DROP [313:33120]
:FORWARD DROP [9:756]
:OUTPUT DROP [0:0]
-A INPUT -s 192.168.2.0/255.255.255.0 -d 192.168.2.1 -i eth1 -m state --state NEW -j
ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -j LOG --log-prefix "iptables INPUT: " --log-level 6
-A FORWARD -s 192.168.2.0/255.255.255.0 -m state --state NEW -j ACCEPT
-A FORWARD -d 192.168.2.9 -p tcp -m state --state NEW,RELATED,ESTABLISHED -m tcp --dport
23 -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -j LOG --log-prefix "iptables FORWARD: " --log-level 6
-A OUTPUT -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
[root@elrond sysconfig]#
```

Firewall and NAT settings for Lab 5

Elrond

Standard NAT chains

```
[root@elrond sysconfig]# cat iptables.lab5
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*nat
:PREROUTING ACCEPT [376:36875]
:POSTROUTING ACCEPT [74:4747]
:OUTPUT ACCEPT [60:3780]
-A PREROUTING -d 172.30.4.122 -i eth0 -j DNAT --to-destination 192.168.2.9
-A POSTROUTING -s 192.168.2.9 -o eth0 -j SNAT --to-source 172.30.4.122
-A POSTROUTING -s 192.168.2.0/255.255.255.0 -o eth0 -j SNAT --to-source 172.30.4.121
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*filter
:INPUT DROP [313:33120]
:FORWARD DROP [9:756]
:OUTPUT DROP [0:0]
-A INPUT -s 192.168.2.0/255.255.255.0 -d 192.168.2.1 -i eth1 -m state --state NEW -j
ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -j LOG --log-prefix "iptables INPUT: " --log-level 6
-A FORWARD -s 192.168.2.0/255.255.255.0 -m state --state NEW -j ACCEPT
-A FORWARD -d 192.168.2.9 -p tcp -m state --state NEW,RELATED,ESTABLISHED -m tcp --dport
23 -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -j LOG --log-prefix "iptables FORWARD: " --log-level 6
-A OUTPUT -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
[root@elrond sysconfig]#
```

Firewall and NAT settings for Lab 5

Elrond

```
[root@elrond sysconfig]# cat iptables.lab5
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*nat
:PREROUTING ACCEPT [376:36875]
:POSTROUTING ACCEPT [74:4747]
:OUTPUT ACCEPT [60:3780]
-A PREROUTING -d 172.30.4.122 -i eth0 -j DNAT --to-destination 192.168.2.9
-A POSTROUTING -s 192.168.2.9 -o eth0 -j SNAT --to-source 172.30.4.122
-A POSTROUTING -s 192.168.2.0/255.255.255.0 -o eth0 -j SNAT --to-source 172.30.4.121
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*filter
:INPUT DROP [313:33120]
:FORWARD DROP [9:756]
:OUTPUT DROP [0:0]
-A INPUT -s 192.168.2.0/255.255.255.0 -d 192.168.2.1 -i eth1 -m state --state NEW -j
ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -j LOG --log-prefix "iptables INPUT: " --log-level 6
-A FORWARD -s 192.168.2.0/255.255.255.0 -m state --state NEW -j ACCEPT
-A FORWARD -d 192.168.2.9 -p tcp -m state --state NEW,RELATED,ESTABLISHED -m tcp --dport
23 -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -j LOG --log-prefix "iptables FORWARD: " --log-level 6
-A OUTPUT -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
[root@elrond sysconfig]#
```

Standard filter chains

Using lab addresses for this example

Firewall and NAT settings for Lab 5

Elrond

Policy settings which are used if no rules on the chain apply

```
[root@elrond sysconfig]# cat iptables.lab5
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*nat
:PREROUTING ACCEPT [376:36875]
:POSTROUTING ACCEPT [74:4747]
:OUTPUT ACCEPT [60:3780]
-A PREROUTING -d 172.30.4.122 -i eth0 -j DNAT --to-destination 192.168.2.9
-A POSTROUTING -s 192.168.2.9 -o eth0 -j SNAT --to-source 172.30.4.122
-A POSTROUTING -s 192.168.2.0/255.255.255.0 -o eth0 -j SNAT --to-source 172.30.4.121
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*filter
:INPUT DROP [313:33120]
:FORWARD DROP [9:756]
:OUTPUT DROP [0:0]
-A INPUT -s 192.168.2.0/255.255.255.0 -d 192.168.2.1 -i eth1 -m state --state NEW -j
ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -j LOG --log-prefix "iptables INPUT: " --log-level 6
-A FORWARD -s 192.168.2.0/255.255.255.0 -m state --state NEW -j ACCEPT
-A FORWARD -d 192.168.2.9 -p tcp -m state --state NEW,RELATED,ESTABLISHED -m tcp --dport
23 -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -j LOG --log-prefix "iptables FORWARD: " --log-level 6
-A OUTPUT -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
[root@elrond sysconfig]#
```

Firewall and NAT settings for Lab 5

Elrond

Forward any packets to 172.30.4.122 to 192.168.2.9

```
[root@elrond sysconfig]# cat iptables.lab5
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*nat
:PREROUTING ACCEPT [376:36875]
:POSTROUTING ACCEPT [74:4747]
:OUTPUT ACCEPT [60:3780]
-A PREROUTING -d 172.30.4.122 -i eth0 -j DNAT --to-destination 192.168.2.9
-A POSTROUTING -s 192.168.2.9 -o eth0 -j SNAT --to-source 172.30.4.122
-A POSTROUTING -s 192.168.2.0/255.255.255.0 -o eth0 -j SNAT --to-source 172.30.4.121
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*filter
:INPUT DROP [313:33120]
:FORWARD DROP [9:756]
:OUTPUT DROP [0:0]
-A INPUT -s 192.168.2.0/255.255.255.0 -d 192.168.2.1 -i eth1 -m state --state NEW -j
ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -j LOG --log-prefix "iptables INPUT: " --log-level 6
-A FORWARD -s 192.168.2.0/255.255.255.0 -m state --state NEW -j ACCEPT
-A FORWARD -d 192.168.2.9 -p tcp -m state --state NEW,RELATED,ESTABLISHED -m tcp --dport
23 -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -j LOG --log-prefix "iptables FORWARD: " --log-level 6
-A OUTPUT -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
[root@elrond sysconfig]#
```

Firewall and NAT settings for Lab 5

Elrond

Make outgoing packets from 192.168.2.9 appear as if they came from 172.30.4.122 (NAT)

```
[root@elrond sysconfig]# cat iptables.lab5
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*nat
:PREROUTING ACCEPT [376:36875]
:POSTROUTING ACCEPT [74:4747]
:OUTPUT ACCEPT [60:3780]
-A PREROUTING -d 172.30.4.122 -i eth0 -j DNAT --to-destination 192.168.2.9
-A POSTROUTING -s 192.168.2.9 -o eth0 -j SNAT --to-source 172.30.4.122
-A POSTROUTING -s 192.168.2.0/255.255.255.0 -o eth0 -j SNAT --to-source 172.30.4.121
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*filter
:INPUT DROP [313:33120]
:FORWARD DROP [9:756]
:OUTPUT DROP [0:0]
-A INPUT -s 192.168.2.0/255.255.255.0 -d 192.168.2.1 -i eth1 -m state --state NEW -j
ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -j LOG --log-prefix "iptables INPUT: " --log-level 6
-A FORWARD -s 192.168.2.0/255.255.255.0 -m state --state NEW -j ACCEPT
-A FORWARD -d 192.168.2.9 -p tcp -m state --state NEW,RELATED,ESTABLISHED -m tcp --dport
23 -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -j LOG --log-prefix "iptables FORWARD: " --log-level 6
-A OUTPUT -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
[root@elrond sysconfig]#
```

Firewall and NAT settings for Lab 5

Elrond

Make outgoing packets from private 192.168.2.0/24 network appear as if they came from 172.30.4.121 (NAT)

```
[root@elrond sysconfig]# cat iptables.lab5
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*nat
:PREROUTING ACCEPT [376:36875]
:POSTROUTING ACCEPT [74:4747]
:OUTPUT ACCEPT [60:3780]
-A PREROUTING -d 172.30.4.122 -i eth0 -j DNAT --to-destination 192.168.2.9
-A POSTROUTING -s 192.168.2.9 -o eth0 -j SNAT --to-source 172.30.4.122
-A POSTROUTING -s 192.168.2.0/255.255.255.0 -o eth0 -j SNAT --to-source 172.30.4.121
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*filter
:INPUT DROP [313:33120]
:FORWARD DROP [9:756]
:OUTPUT DROP [0:0]
-A INPUT -s 192.168.2.0/255.255.255.0 -d 192.168.2.1 -i eth1 -m state --state NEW -j
ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -j LOG --log-prefix "iptables INPUT: " --log-level 6
-A FORWARD -s 192.168.2.0/255.255.255.0 -m state --state NEW -j ACCEPT
-A FORWARD -d 192.168.2.9 -p tcp -m state --state NEW,RELATED,ESTABLISHED -m tcp --dport
23 -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -j LOG --log-prefix "iptables FORWARD: " --log-level 6
-A OUTPUT -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
[root@elrond sysconfig]#
```

Firewall and NAT settings for Lab 5

Elrond

```
[root@elrond sysconfig]# cat iptables.lab5
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*nat
:PREROUTING ACCEPT [376:36875]
:POSTROUTING ACCEPT [74:4747]
:OUTPUT ACCEPT [60:3780]
-A PREROUTING -d 172.30.4.122 -i eth0 -j DNAT --to-destination 192.168.2.9
-A POSTROUTING -s 192.168.2.9 -o eth0 -j SNAT --to-source 172.30.4.122
-A POSTROUTING -s 192.168.2.0/255.255.255.0 -o eth0 -j SNAT --to-source 172.30.4.121
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*filter
:INPUT DROP [313:33120]
:FORWARD DROP [9:756]
:OUTPUT DROP [0:0]
-A INPUT -s 192.168.2.0/255.255.255.0 -d 192.168.2.1 -i eth1 -m state --state NEW -j
ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -j LOG --log-prefix "iptables INPUT: " --log-level 6
-A FORWARD -s 192.168.2.0/255.255.255.0 -m state --state NEW -j ACCEPT
-A FORWARD -d 192.168.2.9 -p tcp -m state --state NEW,RELATED,ESTABLISHED -m tcp --dport
23 -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -j LOG --log-prefix "iptables FORWARD: " --log-level 6
-A OUTPUT -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
[root@elrond sysconfig]#
```

Allow incoming ongoing traffic based on previous new connections that were allowed

Firewall and NAT settings for Lab 5

Elrond

```
[root@elrond sysconfig]# cat iptables.lab5
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*nat
:PREROUTING ACCEPT [376:36875]
:POSTROUTING ACCEPT [74:4747]
:OUTPUT ACCEPT [60:3780]
-A PREROUTING -d 172.30.4.122 -i eth0 -j DNAT --to-destination 192.168.2.9
-A POSTROUTING -s 192.168.2.9 -o eth0 -j SNAT --to-source 172.30.4.122
-A POSTROUTING -s 192.168.2.0/255.255.255.0 -o eth0 -j SNAT --to-source 172.30.4.121
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*filter
:INPUT DROP [313:33120]
:FORWARD DROP [9:756]
:OUTPUT DROP [0:0]
-A INPUT -s 192.168.2.0/255.255.255.0 -d 192.168.2.1 -i eth1 -m state --state NEW -j
ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -j LOG --log-prefix "iptables INPUT: " --log-level 6
-A FORWARD -s 192.168.2.0/255.255.255.0 -m state --state NEW -j ACCEPT
-A FORWARD -d 192.168.2.9 -p tcp -m state --state NEW,RELATED,ESTABLISHED -m tcp --dport
23 -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -j LOG --log-prefix "iptables FORWARD: " --log-level 6
-A OUTPUT -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
[root@elrond sysconfig]#
```

Log any input traffic that was not filtered out by rules above

Firewall and NAT settings for Lab 5

Elrond

```
[root@elrond sysconfig]# cat iptables.lab5
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*nat
:PREROUTING ACCEPT [376:36875]
:POSTROUTING ACCEPT [74:4747]
:OUTPUT ACCEPT [60:3780]
-A PREROUTING -d 172.30.4.122 -i eth0 -j DNAT --to-destination 192.168.2.9
-A POSTROUTING -s 192.168.2.9 -o eth0 -j SNAT --to-source 172.30.4.122
-A POSTROUTING -s 192.168.2.0/255.255.255.0 -o eth0 -j SNAT --to-source 172.30.4.121
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*filter
:INPUT DROP [313:33120]
:FORWARD DROP [9:756]
:OUTPUT DROP [0:0]
-A INPUT -s 192.168.2.0/255.255.255.0 -d 192.168.2.1 -i eth1 -m state --state NEW -j
ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -j LOG --log-prefix "iptables INPUT: " --log-level 6
-A FORWARD -s 192.168.2.0/255.255.255.0 -m state --state NEW -j ACCEPT
-A FORWARD -d 192.168.2.9 -p tcp -m state --state NEW,RELATED,ESTABLISHED -m tcp --dport
23 -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -j LOG --log-prefix "iptables FORWARD: " --log-level 6
-A OUTPUT -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
[root@elrond sysconfig]#
```

Forward all new outgoing connections from hosts on the private network

Firewall and NAT settings for Lab 5

Elrond

```
[root@elrond sysconfig]# cat iptables.lab5
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*nat
:PREROUTING ACCEPT [376:36875]
:POSTROUTING ACCEPT [74:4747]
:OUTPUT ACCEPT [60:3780]
-A PREROUTING -d 172.30.4.122 -i eth0 -j DNAT --to-destination 192.168.2.9
-A POSTROUTING -s 192.168.2.9 -o eth0 -j SNAT --to-source 172.30.4.122
-A POSTROUTING -s 192.168.2.0/255.255.255.0 -o eth0 -j SNAT --to-source 172.30.4.121
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*filter
:INPUT DROP [313:33120]
:FORWARD DROP [9:756]
:OUTPUT DROP [0:0]
-A INPUT -s 192.168.2.0/255.255.255.0 -d 192.168.2.1 -i eth1 -m state --state NEW -j
ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -j LOG --log-prefix "iptables INPUT: " --log-level 6
-A FORWARD -s 192.168.2.0/255.255.255.0 -m state --state NEW -j ACCEPT
-A FORWARD -d 192.168.2.9 -p tcp -m state --state NEW,RELATED,ESTABLISHED -m tcp --dport
23 -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -j LOG --log-prefix "iptables FORWARD: " --log-level 6
-A OUTPUT -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
[root@elrond sysconfig]#
```

Forward all traffic going to 192.168.2.9 port 23 (the Telnet server)

Firewall and NAT settings for Lab 5

Elrond

```
[root@elrond sysconfig]# cat iptables.lab5
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*nat
:PREROUTING ACCEPT [376:36875]
:POSTROUTING ACCEPT [74:4747]
:OUTPUT ACCEPT [60:3780]
-A PREROUTING -d 172.30.4.122 -i eth0 -j DNAT --to-destination 192.168.2.9
-A POSTROUTING -s 192.168.2.9 -o eth0 -j SNAT --to-source 172.30.4.122
-A POSTROUTING -s 192.168.2.0/255.255.255.0 -o eth0 -j SNAT --to-source 172.30.4.121
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*filter
:INPUT DROP [313:33120]
:FORWARD DROP [9:756]
:OUTPUT DROP [0:0]
-A INPUT -s 192.168.2.0/255.255.255.0 -d 192.168.2.1 -i eth1 -m state --state NEW -j
ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -j LOG --log-prefix "iptables INPUT: " --log-level 6
-A FORWARD -s 192.168.2.0/255.255.255.0 -m state --state NEW -j ACCEPT
-A FORWARD -d 192.168.2.9 -p tcp -m state --state NEW,RELATED,ESTABLISHED -m tcp --dport
23 -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -j LOG --log-prefix "iptables FORWARD: " --log-level 6
-A OUTPUT -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
[root@elrond sysconfig]#
```

Forward ongoing traffic based on previous new connections that were allowed

Firewall and NAT settings for Lab 5

Elrond

```
[root@elrond sysconfig]# cat iptables.lab5
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*nat
:PREROUTING ACCEPT [376:36875]
:POSTROUTING ACCEPT [74:4747]
:OUTPUT ACCEPT [60:3780]
-A PREROUTING -d 172.30.4.122 -i eth0 -j DNAT --to-destination 192.168.2.9
-A POSTROUTING -s 192.168.2.9 -o eth0 -j SNAT --to-source 172.30.4.122
-A POSTROUTING -s 192.168.2.0/255.255.255.0 -o eth0 -j SNAT --to-source 172.30.4.121
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*filter
:INPUT DROP [313:33120]
:FORWARD DROP [9:756]
:OUTPUT DROP [0:0]
-A INPUT -s 192.168.2.0/255.255.255.0 -d 192.168.2.1 -i eth1 -m state --state NEW -j
ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -j LOG --log-prefix "iptables INPUT: " --log-level 6
-A FORWARD -s 192.168.2.0/255.255.255.0 -m state --state NEW -j ACCEPT
-A FORWARD -d 192.168.2.9 -p tcp -m state --state NEW,RELATED,ESTABLISHED -m tcp --dport
23 -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -j LOG --log-prefix "iptables FORWARD: " --log-level 6
-A OUTPUT -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
[root@elrond sysconfig]#
```

Log any traffic that is about to be dropped (this is the last rule on the chain before policy get applied)

Firewall and NAT settings for Lab 5

Elrond

```
[root@elrond sysconfig]# cat iptables.lab5
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*nat
:PREROUTING ACCEPT [376:36875]
:POSTROUTING ACCEPT [74:4747]
:OUTPUT ACCEPT [60:3780]
-A PREROUTING -d 172.30.4.122 -i eth0 -j DNAT --to-destination 192.168.2.9
-A POSTROUTING -s 192.168.2.9 -o eth0 -j SNAT --to-source 172.30.4.122
-A POSTROUTING -s 192.168.2.0/255.255.255.0 -o eth0 -j SNAT --to-source 172.30.4.121
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
# Generated by iptables-save v1.3.5 on Sun Mar 14 16:08:44 2010
*filter
:INPUT DROP [313:33120]
:FORWARD DROP [9:756]
:OUTPUT DROP [0:0]
-A INPUT -s 192.168.2.0/255.255.255.0 -d 192.168.2.1 -i eth1 -m state --state NEW -j
ACCEPT
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -j LOG --log-prefix "iptables INPUT: " --log-level 6
-A FORWARD -s 192.168.2.0/255.255.255.0 -m state --state NEW -j ACCEPT
-A FORWARD -d 192.168.2.9 -p tcp -m state --state NEW,RELATED,ESTABLISHED -m tcp --dport
23 -j ACCEPT
-A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -j LOG --log-prefix "iptables FORWARD: " --log-level 6
-A OUTPUT -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
COMMIT
# Completed on Sun Mar 14 16:08:44 2010
[root@elrond sysconfig]#
```

Allow all outgoing traffic

The Nosmo VM


```
[root@nosmo root]# cat /etc/sysconfig/iptables
# Generated by iptables-save v1.2.7a on Mon Jan 11 12:14:00 2010
*filter
:INPUT ACCEPT [4229:434875]
:FORWARD ACCEPT [1481:444016]
:OUTPUT ACCEPT [3340:350240]
COMMIT
# Completed on Mon Jan 11 12:14:00 2010
# Generated by iptables-save v1.2.7a on Mon Jan 11 12:14:00 2010
*nat
:PREROUTING ACCEPT [8414:1265541]
:POSTROUTING ACCEPT [226:15381]
:OUTPUT ACCEPT [95:7826]
-A PREROUTING -d 207.62.187.53 -j DNAT --to-destination 192.168.0.1
-A POSTROUTING -o eth0 -j MASQUERADE
COMMIT
# Completed on Mon Jan 11 12:14:00 2010
[root@nosmo root]#
```

This is the DNS server IP address I use at home on my Netgear router

Forward DNS traffic intended for Bubbles (Cabrillo DNS server) to DNS server use at home

The Nosmo VM


```
[root@nosmo root]# cat /etc/sysconfig/iptables
# Generated by iptables-save v1.2.7a on Mon Jan 11 12:14:00 2010
*filter
:INPUT ACCEPT [4229:434875]
:FORWARD ACCEPT [1481:444016]
:OUTPUT ACCEPT [3340:350240]
COMMIT
# Completed on Mon Jan 11 12:14:00 2010
# Generated by iptables-save v1.2.7a on Mon Jan 11 12:14:00 2010
*nat
:PREROUTING ACCEPT [8414:1265541]
:POSTROUTING ACCEPT [226:15381]
:OUTPUT ACCEPT [95:7826]
-A PREROUTING -d 207.62.187.53 -j DNAT --to-destination 192.168.0.1
-A POSTROUTING -o eth0 -j MASQUERADE
COMMIT
# Completed on Mon Jan 11 12:14:00 2010
[root@nosmo root]#
```

NAT all outgoing traffic to the public IP address on Nosmo.
This give Shire hosts Internet access

Impeding brute force attacks

Recent dictionary attack from 202.113.16.118

```
[root@opus ~]# lastb | grep 202.113.16.118
recruit  ssh:notty    202.113.16.118   Sun Mar 14 11:32 - 11:32  (00:00)
recruit  ssh:notty    202.113.16.118   Sun Mar 14 11:32 - 11:32  (00:00)
sales    ssh:notty    202.113.16.118   Sun Mar 14 11:32 - 11:32  (00:00)
sales    ssh:notty    202.113.16.118   Sun Mar 14 11:32 - 11:32  (00:00)
staff    ssh:notty    202.113.16.118   Sun Mar 14 11:32 - 11:32  (00:00)
staff    ssh:notty    202.113.16.118   Sun Mar 14 11:32 - 11:32  (00:00)
[root@opus ~]#
```

Impeding brute force attacks

Adds the current IP address to the recent list using the recent module

```
[rsimms@opus ~]$ cat /etc/sysconfig/iptables
< snipped >
# Impede brute force SSH dictionary attacks using the recent module (Rule added by RJS)
-A RH-Firewall-1-INPUT -p tcp --dport 22 -m state --state NEW -m recent --set –name SHBF
-A RH-Firewall-1-INPUT -p tcp --dport 22 -m state --state NEW -m recent --update --
seconds 60 --hitcount 4 --rttl --name SSHBF -j LOG --log-level info --log-prefix
"iptables brute force block: "
-A RH-Firewall-1-INPUT -p tcp --dport 22 -m state --state NEW -m recent --update --
seconds 60 --hitcount 4 --rttl --name SSHBF -j DROP
< snipped >
[rsimms@opus ~]$
```

http://kevin.vanzonneveld.net/techblog/article/block_brute_force_attacks_with_iptables/

Impeding brute force attacks

If four packets were sent from the same IP address in the last 60 seconds then log the packet.

```
[rsimms@opus ~]$ cat /etc/sysconfig/iptables
< snipped >
# Impede brute force SSH dictionary attacks using the recent module (Rule added by RJS)
-A RH-Firewall-1-INPUT -p tcp --dport 22 -m state --state NEW -m recent --set –name SHBF
-A RH-Firewall-1-INPUT -p tcp --dport 22 -m state --state NEW -m recent --update --
seconds 60 --hitcount 4 --rttl --name SSHBF -j LOG --log-level info --log-prefix
"iptables brute force block: "
-A RH-Firewall-1-INPUT -p tcp --dport 22 -m state --state NEW -m recent --update --
seconds 60 --hitcount 4 --rttl --name SSHBF -j DROP
< snipped >
[rsimms@opus ~]$
```

Impeding brute force attacks

If four packets were sent from the same IP address in the last 60 seconds then drop the packet.

```
[rsimms@opus ~]$ cat /etc/sysconfig/iptables
< snipped >
# Impede brute force SSH dictionary attacks using the recent module (Rule added by RJS)
-A RH-Firewall-1-INPUT -p tcp --dport 22 -m state --state NEW -m recent --set –name SHBF
-A RH-Firewall-1-INPUT -p tcp --dport 22 -m state --state NEW -m recent --update --
seconds 60 --hitcount 4 --rttl --name SSHBF -j LOG --log-level info --log-prefix
"iptables brute force block: "
-A RH-Firewall-1-INPUT -p tcp --dport 22 -m state --state NEW -m recent --update --
seconds 60 --hitcount 4 --rttl --name SSHBF -j DROP
< snipped >
[rsimms@opus ~]$
```

Impeding brute force attacks

```
[root@opus ~]# cat /var/log/messages | grep brute
< snipped >
Mar 14 11:32:56 Opus kernel: iptables brute force block: IN=eth0 OUT=
MAC=00:50:56:90:7f:d8:00:22:55:97:10:0f:08:00 SRC=202.113.16.118 DST=207.62.186.9 LEN=60
TOS=0x00 PREC=0x00 TTL=49 ID=16335 DF PROTO=TCP SPT=34937 DPT=22 WINDOW=5840 RES=0x00 SYN
URGP=0
Mar 14 11:32:59 Opus kernel: iptables brute force block: IN=eth0 OUT=
MAC=00:50:56:90:7f:d8:00:22:55:97:10:0f:08:00 SRC=202.113.16.118 DST=207.62.186.9 LEN=60
TOS=0x00 PREC=0x00 TTL=49 ID=16336 DF PROTO=TCP SPT=34937 DPT=22 WINDOW=5840 RES=0x00 SYN
URGP=0
Mar 14 11:33:05 Opus kernel: iptables brute force block: IN=eth0 OUT=
MAC=00:50:56:90:7f:d8:00:22:55:97:10:0f:08:00 SRC=202.113.16.118 DST=207.62.186.9 LEN=60
TOS=0x00 PREC=0x00 TTL=49 ID=16337 DF PROTO=TCP SPT=34937 DPT=22 WINDOW=5840 RES=0x00 SYN
URGP=0
Mar 14 13:00:42 Opus kernel: iptables brute force block: IN=eth0 OUT=
MAC=00:50:56:90:7f:d8:00:22:55:97:10:0f:08:00 SRC=121.11.66.70 DST=207.62.186.9 LEN=60
TOS=0x00 PREC=0x00 TTL=50 ID=18877 DF PROTO=TCP SPT=14752 DPT=22 WINDOW=5792 RES=0x00 SYN
URGP=0
Mar 14 13:00:45 Opus kernel: iptables brute force block: IN=eth0 OUT=
MAC=00:50:56:90:7f:d8:00:22:55:97:10:0f:08:00 SRC=121.11.66.70 DST=207.62.186.9 LEN=60
TOS=0x00 PREC=0x00 TTL=50 ID=18879 DF PROTO=TCP SPT=14752 DPT=22 WINDOW=5792 RES=0x00 SYN
URGP=0
Mar 14 13:00:51 Opus kernel: iptables brute force block: IN=eth0 OUT=
MAC=00:50:56:90:7f:d8:00:22:55:97:10:0f:08:00 SRC=121.11.66.70 DST=207.62.186.9 LEN=60
TOS=0x00 PREC=0x00 TTL=50 ID=18881 DF PROTO=TCP SPT=14752 DPT=22 WINDOW=5792 RES=0x00 SYN
URGP=0
Mar 14 16:25:58 Opus kernel: iptables brute force block: IN=eth0 OUT=
< snipped >
```

Netfilter

(iptables)

Netfilter

Netfilter

- Packet filtering (firewall)
- Port and Address translation (NAT*)
- Logging
- Other types of packet mangling
- Implemented by the iptables utility
- Replaces ipchains in older kernels (2.2 and earlier)

*Note, the term NAT can mean different things. Linux really does PAT which includes both address and port translation. This allows multiple private address to be concurrently translated to a single public IP address.

To do this we will use DNAT, SNAT actions or MASQUERADE actions.

Netfilter

Firewalls and Access Control Lists

A Firewall is a system that prevents unauthorized network communications to it, from it, and through it.

Netfilter

IPtables - Chains are grouped into tables:

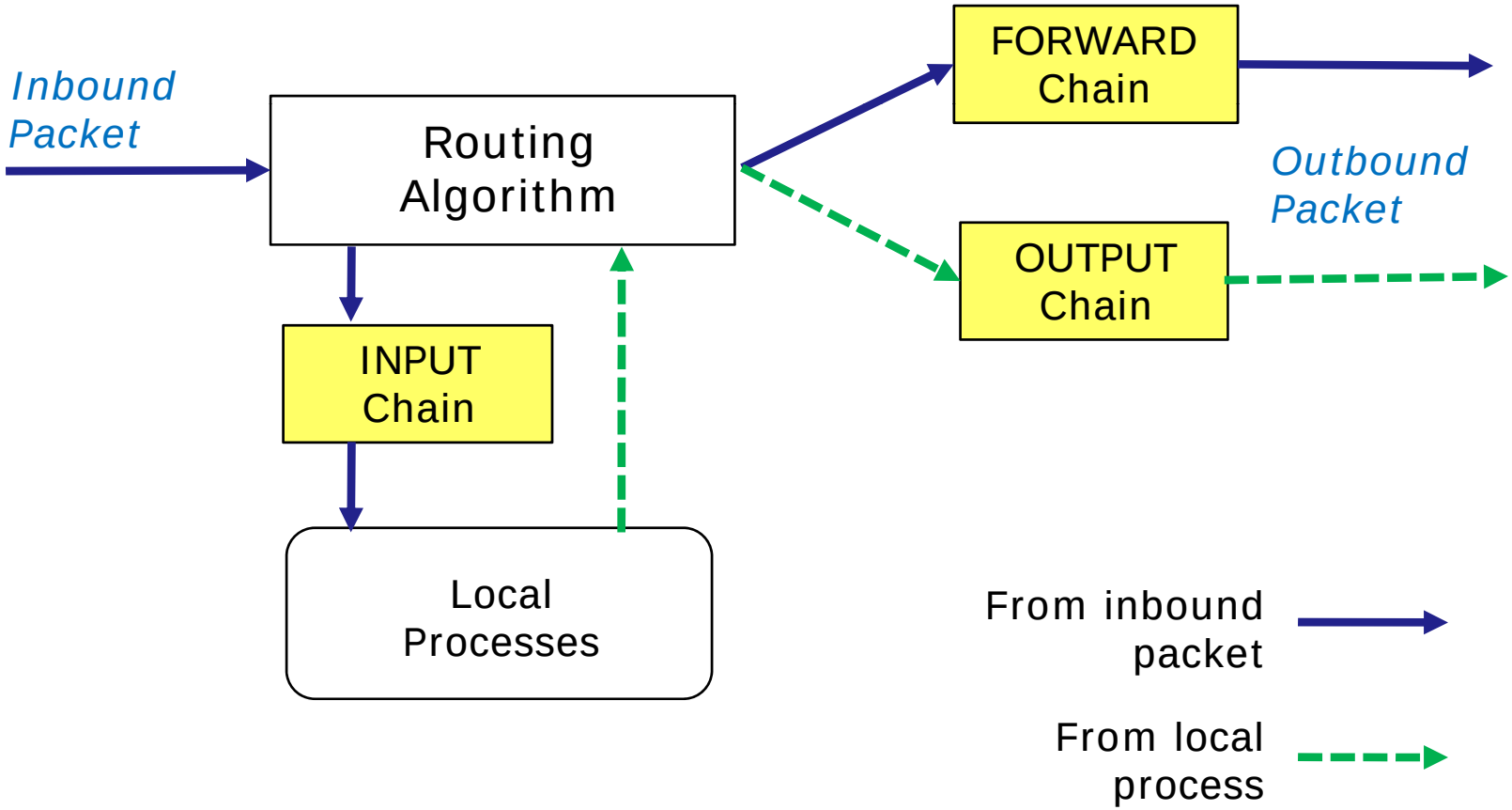
Filter chains:
- Input
- Output
- Forward

NAT chains:
- Prerouting
- Output
- Postrouting

Mangle chains:
- Prerouting
- Output

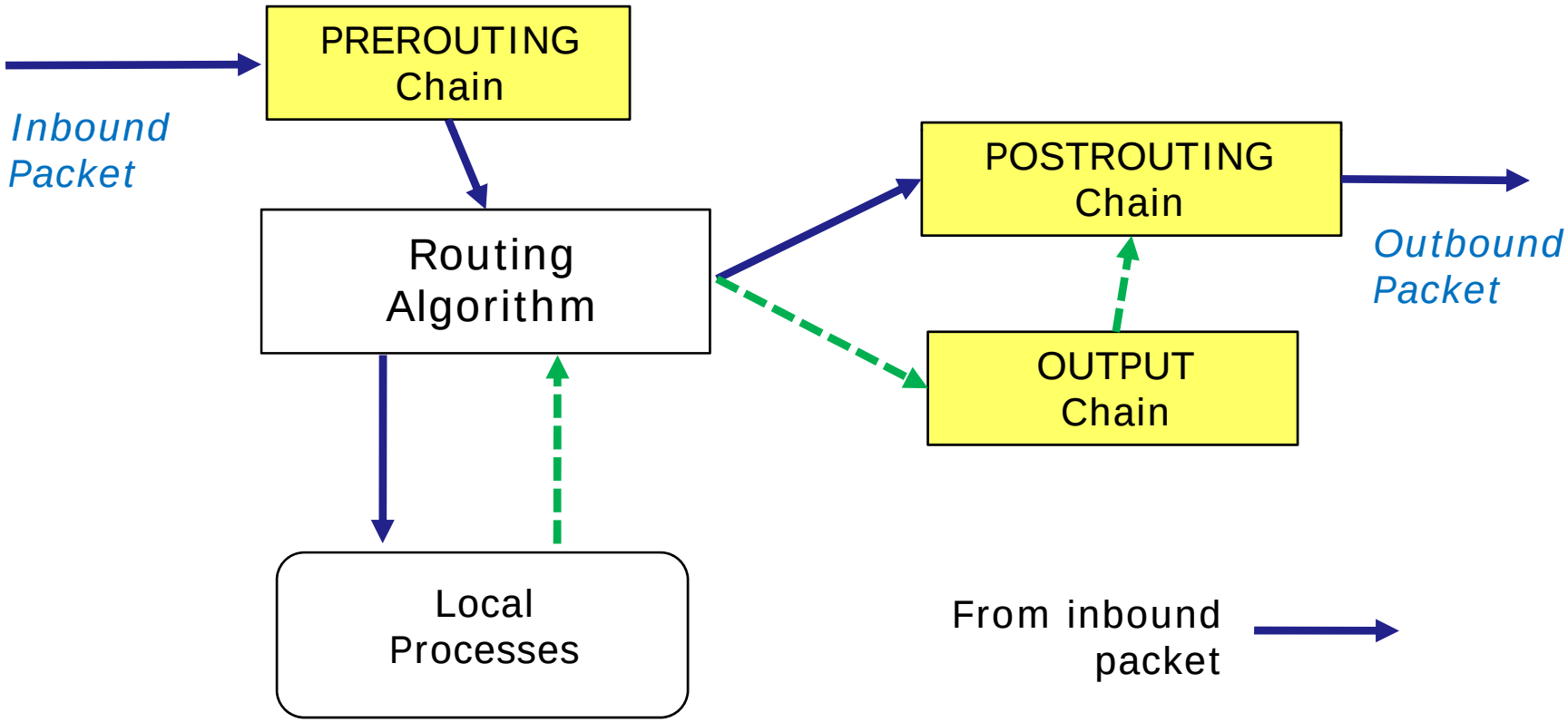
Netfilter

Filter table

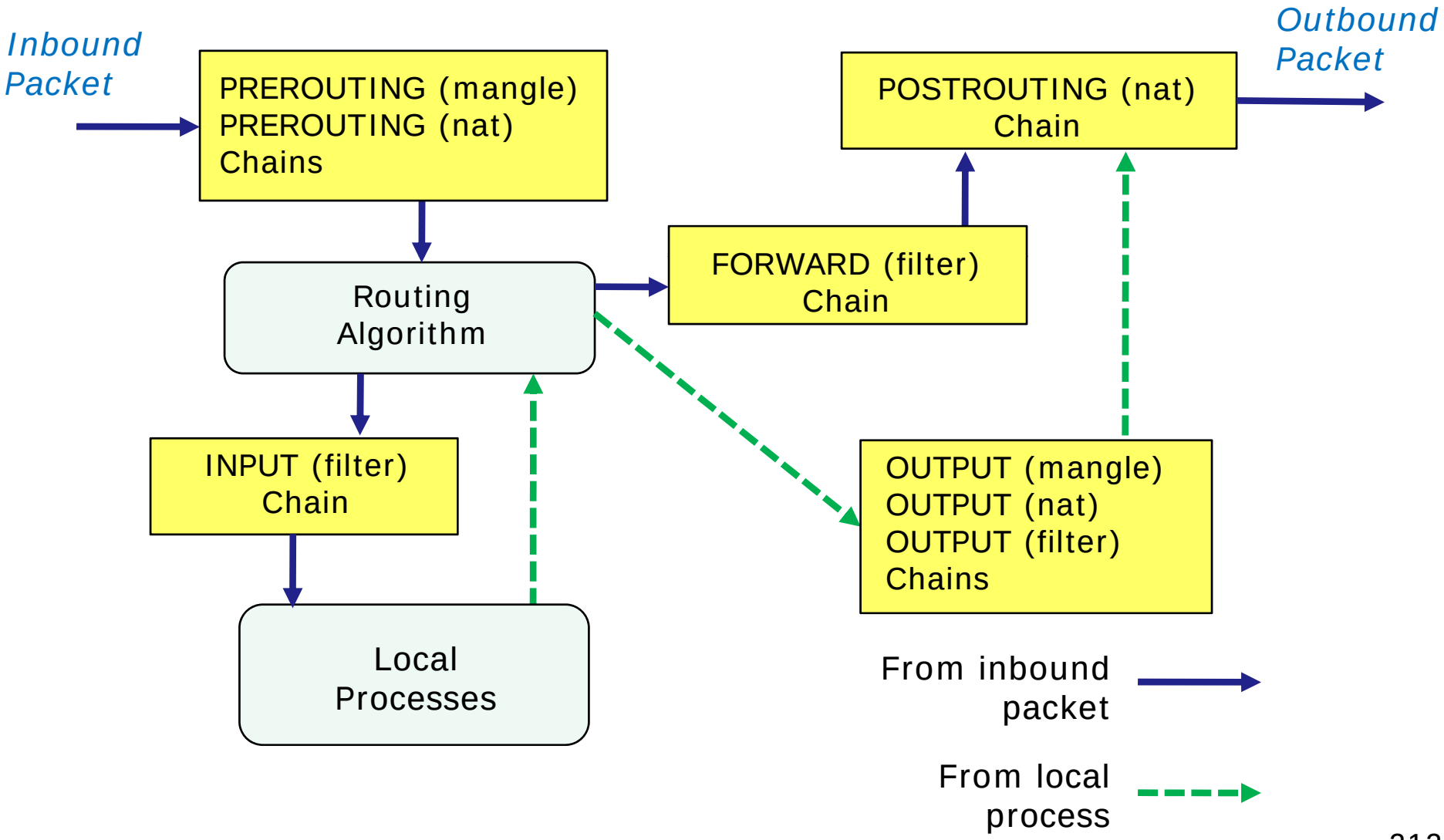


iptables

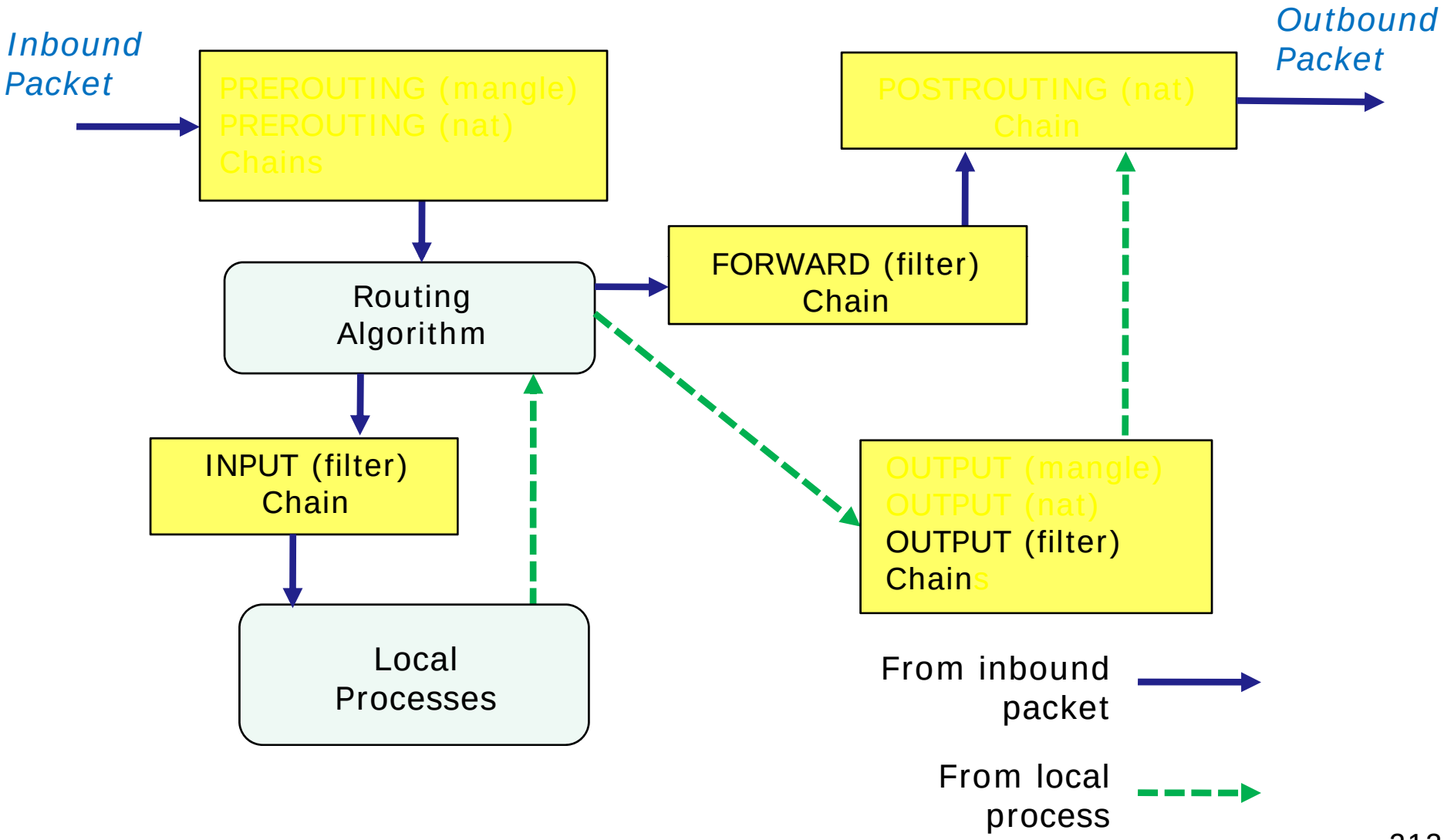
nat table



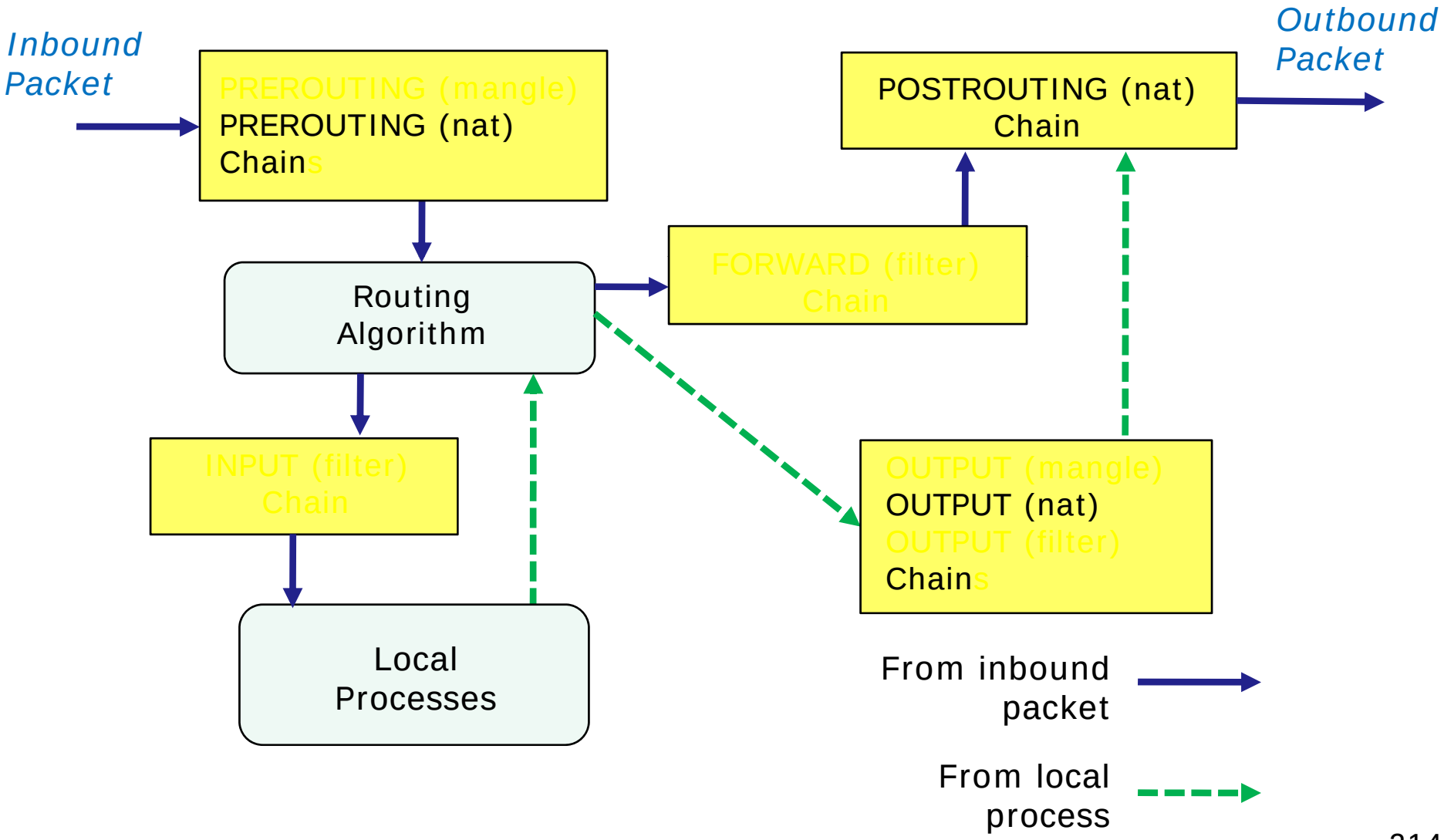
Netfilter – all tables and chains



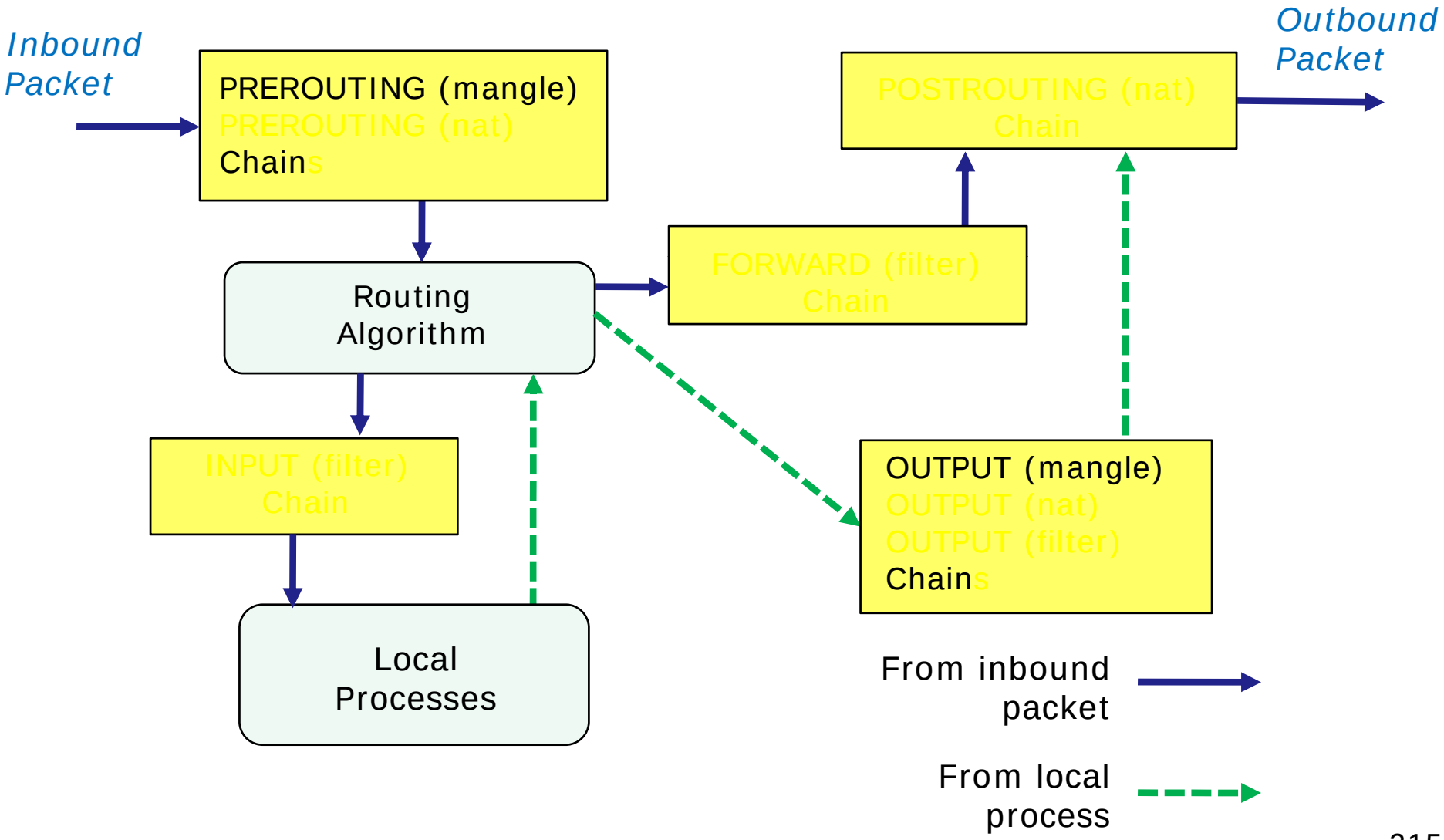
Netfilter – filter table chains



Netfilter – nat table chains



Netfilter – mangle table chains



Netfilter

iptables command syntax

iptables [-flags] [chain] [options [extensions]] [action]

Netfilter

Flags

-t table
-A append a rule
-D delete a rule
-F flush all rules for a specified chain
-I insert a rule at the specified position
-L list all rules
-P policy - the default chain rule
-R replace a rule

Netfilter

Options

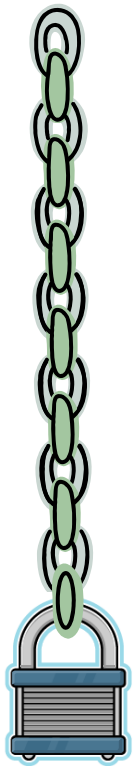
-d destination IP address (accepts CIDR and 0/0 as all)
-s source IP address (accepts CIDR and 0/0 as all)
-p protocol - any name listed in */etc/protocols*
-i the inbound interface
-o the outbound interface
-j the target action
-m extended matching module - has many extensions e.g.
state: --state NEW,ESTABLISHED,RELATED

Netfilter

Actions

ACCEPT
DROP
REJECT
LOG
DNAT
SNAT

Netfilter – chains



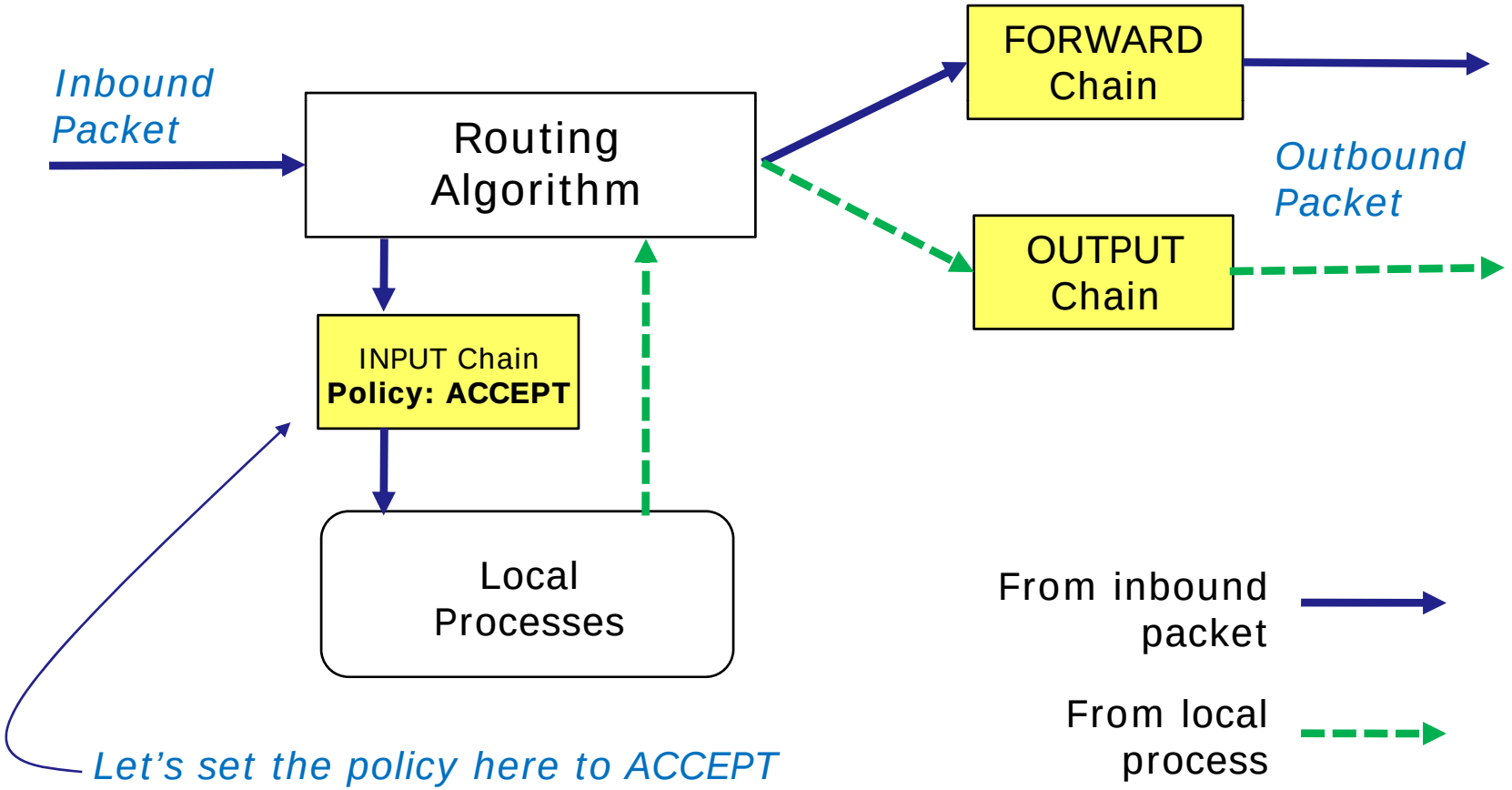
Rules

Policy – the action to take if you get through all the rules on the chain

Table: filter
Chain: INPUT
Policy: ACCEPT
or DROP

Netfilter – examples

Filter table on Elrond



Netfilter – examples

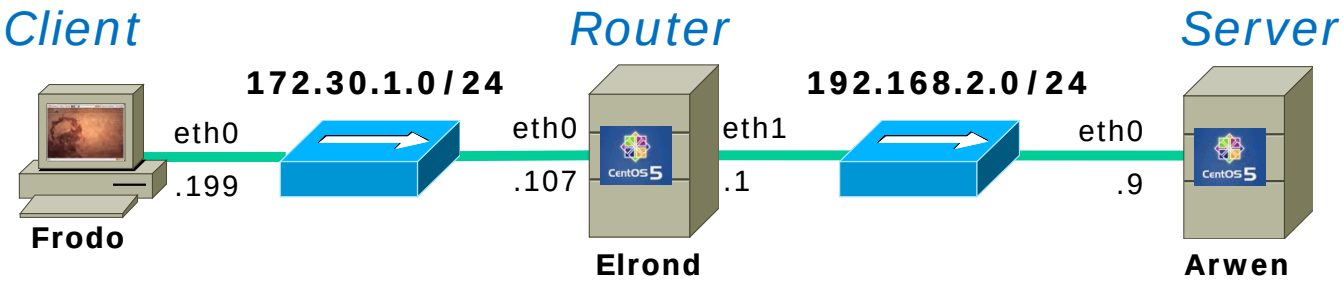
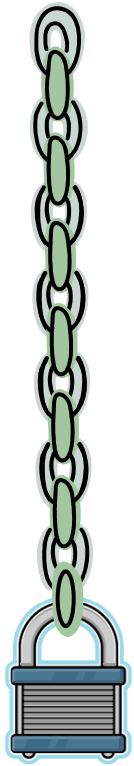


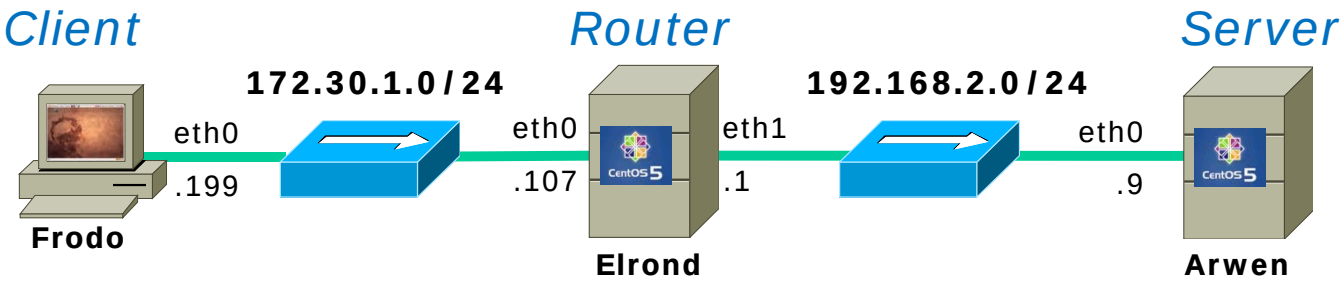
Table: filter
Chain: INPUT



No Rules

Chain Policy: ACCEPT

Netfilter – examples




```
[root@elrond ~]# iptables -F
[root@elrond ~]# iptables -X
[root@elrond ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@elrond ~]#
```

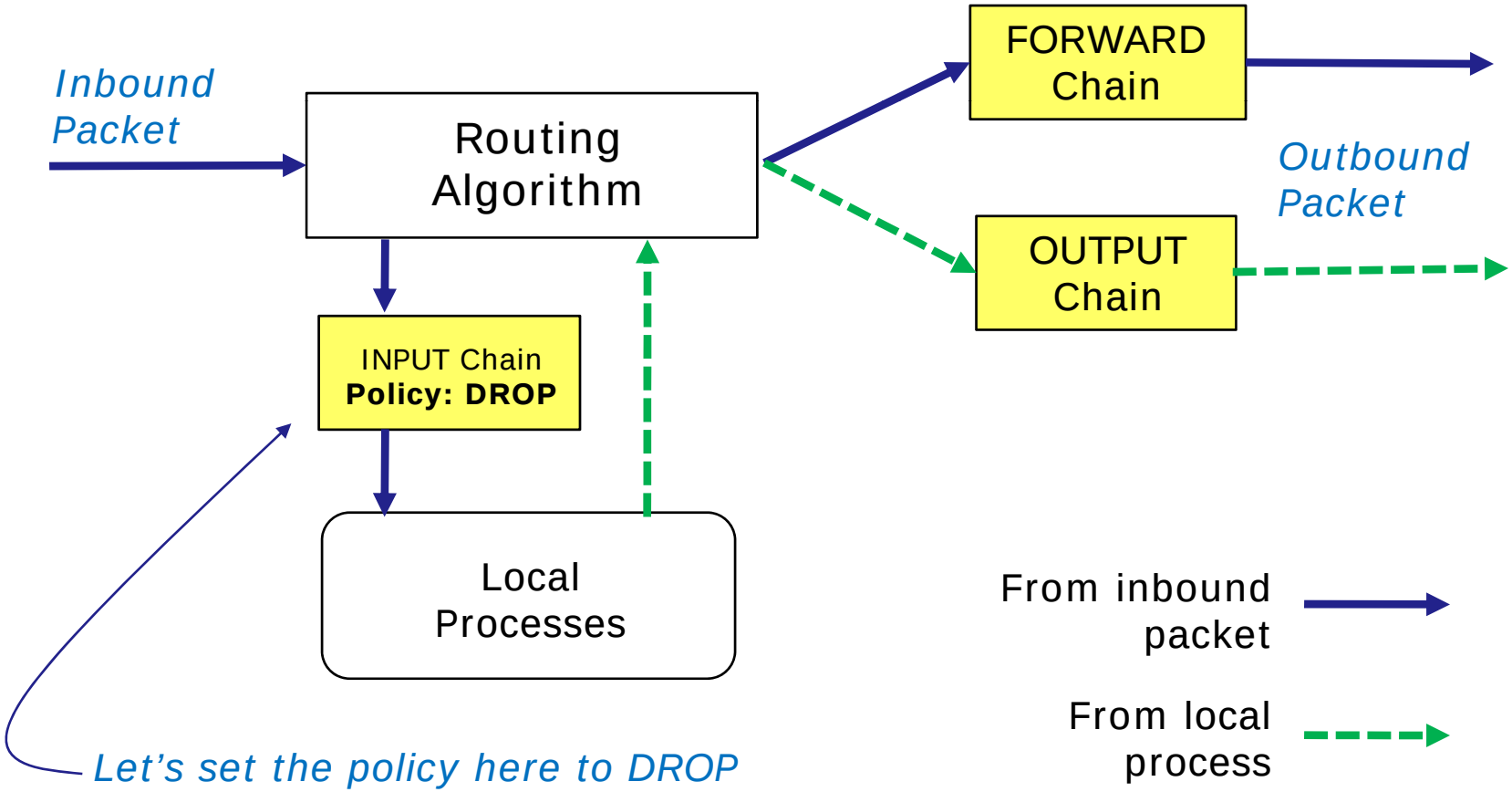
Flush filter chain rules and delete any custom chains.

INPUT chain policy is ACCEPT

```
root@frodo:~# ping -c 1 elrond
PING elrond (172.30.1.107) 56(84) bytes of data.
64 bytes from elrond (172.30.1.107): icmp_seq=1 ttl=64 time=0.803 ms
```

Netfilter – examples

Filter table on Elrond



Netfilter – examples

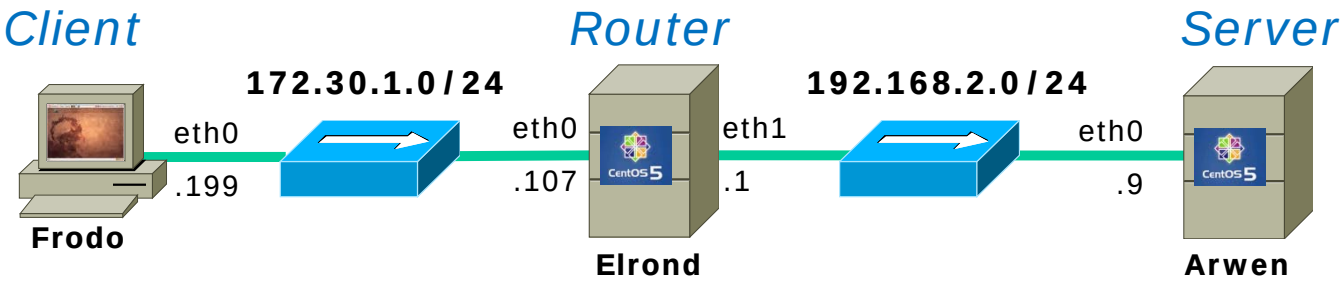
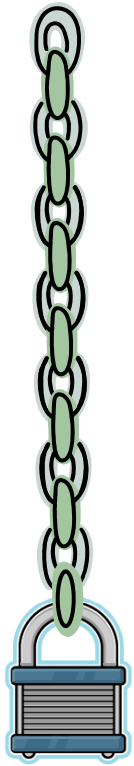


Table: filter
Chain: INPUT

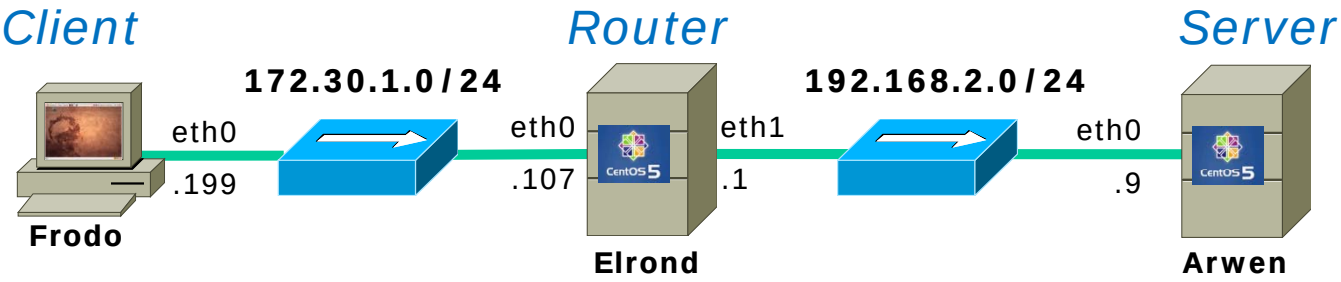


No Rules

Chain Policy: DROP

DROP everything else

Netfilter – examples

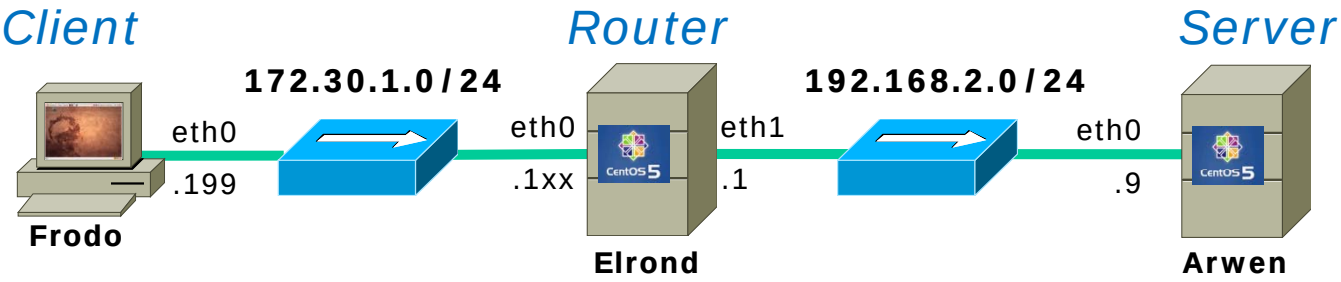



```
[root@elrond ~]# iptables -P INPUT DROP
[root@elrond ~]# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@elrond ~]#
```

Frodo cannot ping Elrond now

```
root@frodo:~# ping -c 2 elrond
PING elrond (172.30.1.107) 56(84) bytes of data.

--- elrond ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 999ms
```

Netfilter – examples

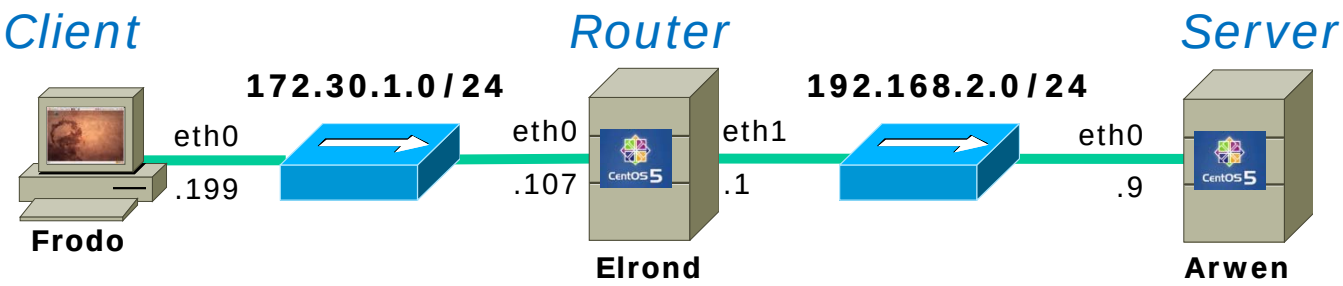


7	3.132262	172.30.4.199	172.30.4.107	ICMP	Echo (ping) request
10	4.132169	172.30.4.199	172.30.4.107	ICMP	Echo (ping) request
11	8.131620	Vmware_6f:53:d9	Vmware_4e:21:af	ARP	Who has 172.30.4.107? Tell 172.30.4.199
12	8.132788	Vmware_4e:21:af	Vmware_6f:53:d9	ARP	172.30.4.107 is at 00:0c:29:4e:21:af
13	10.119859	Vmware_4e:21:af	Vmware_30:16:94	ARP	Who has 172.30.4.1? Tell 172.30.4.107
14	10.119911	Vmware_30:16:94	Vmware_4e:21:af	ARP	172.30.4.1 is at 00:0c:29:30:16:94

Even though Frodo can no longer ping Elrond, Elrond will still respond to Frodo's ARP requests.

```
root@frodo:~# ping -c 2 elrond
PING elrond (172.30.1.107) 56(84) bytes of data.

--- elrond ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 999ms
```

Netfilter – iptables



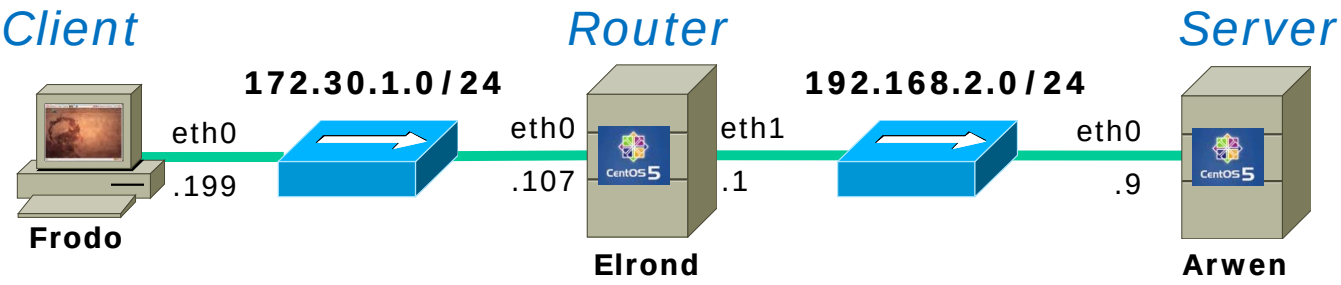

```
[root@elrond ~]# iptables -P INPUT DROP
[root@elrond ~]# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@elrond ~]#
```

Elrond cannot ping Frodo either …

… because the returning echo responses get dropped by the INPUT chain policy

```
[root@elrond ~]# ping -c 2 frodo
PING frodo (172.30.1.199) 56(84) bytes of data.

--- frodo ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1002ms
```

Netfilter – examples



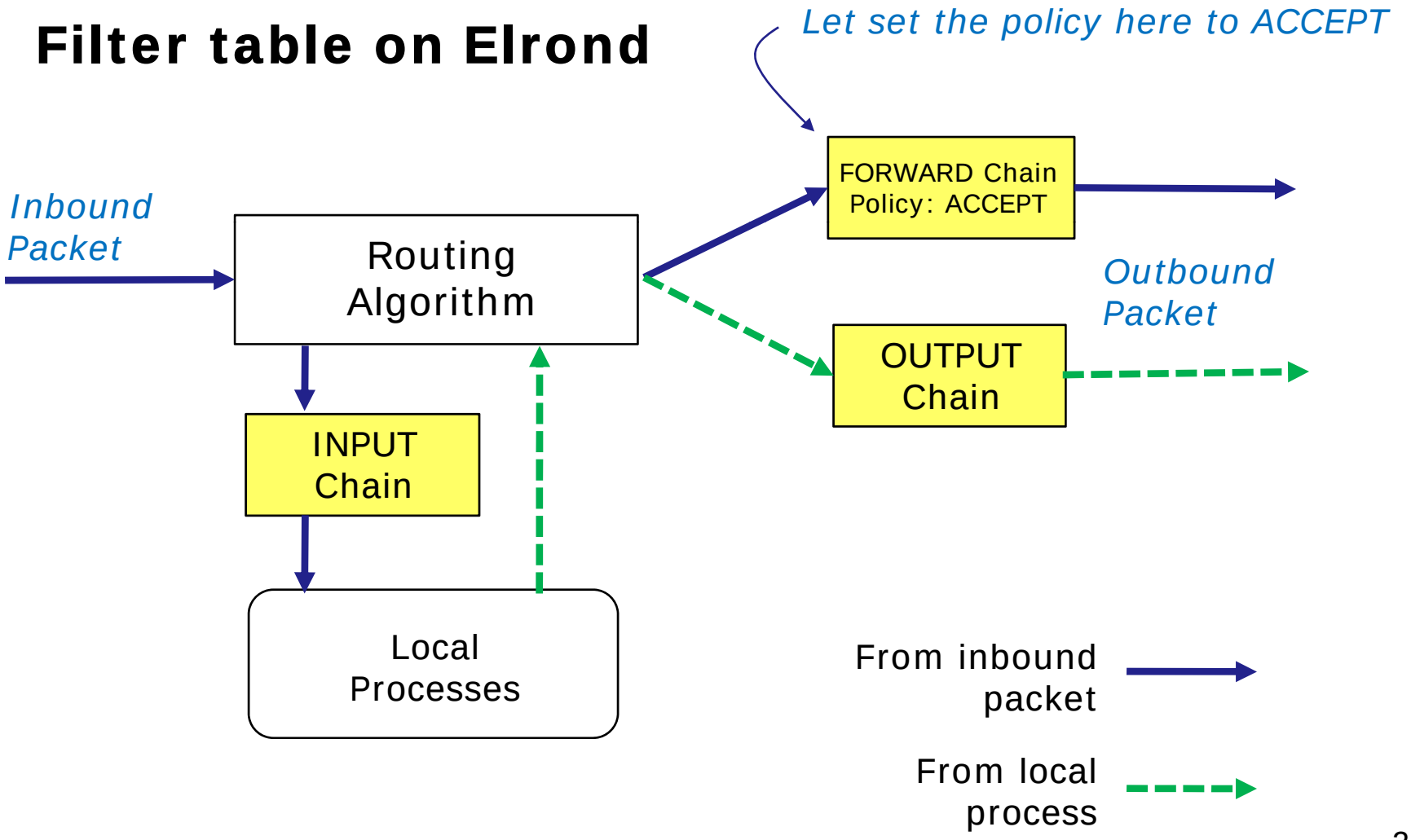
3 2.764346	172.30.4.107	172.30.4.199	ICMP	Echo (ping) request
4 2.764403	172.30.4.199	172.30.4.107	ICMP	Echo (ping) reply
5 4.142478	172.30.4.107	172.30.4.199	ICMP	Echo (ping) request
6 4.143043	172.30.4.199	172.30.4.107	ICMP	Echo (ping) reply
9 7.763088	Vmware_6f:53:d9	Vmware_4e:21:af	ARP	Who has 172.30.4.107? Tell 172.30.4.199
10 7.763496	Vmware_4e:21:af	Vmware_6f:53:d9	ARP	172.30.4.107 is at 00:0c:29:4e:21:af

Note the ping requests get to Frodo and Frodo is responding, however the responses get dropped in Elrond's INPUT chain

```
[root@elrond ~]# ping -c 2 frodo
PING frodo (172.30.1.199) 56(84) bytes of data.

--- frodo ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1002ms
```

Table: filter
Chain: FORWARD
Policy: ACCEPT
or DROP

Netfilter – examples



Netfilter – examples

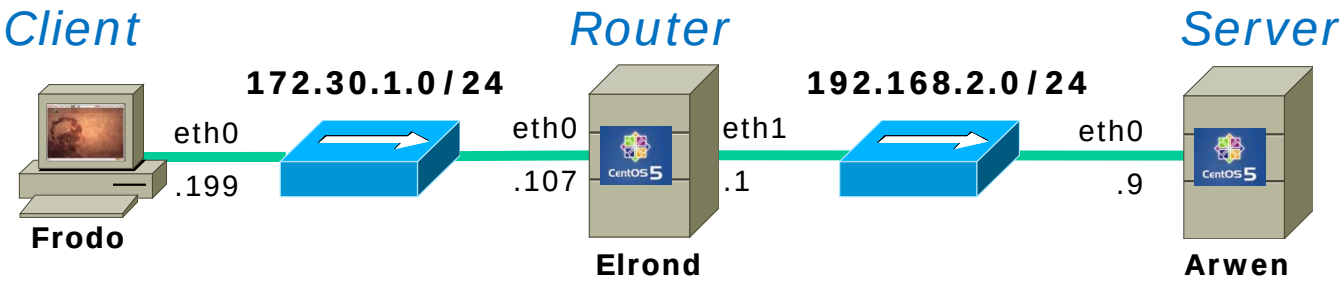
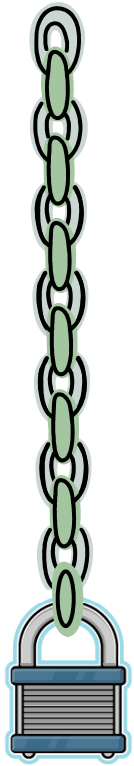


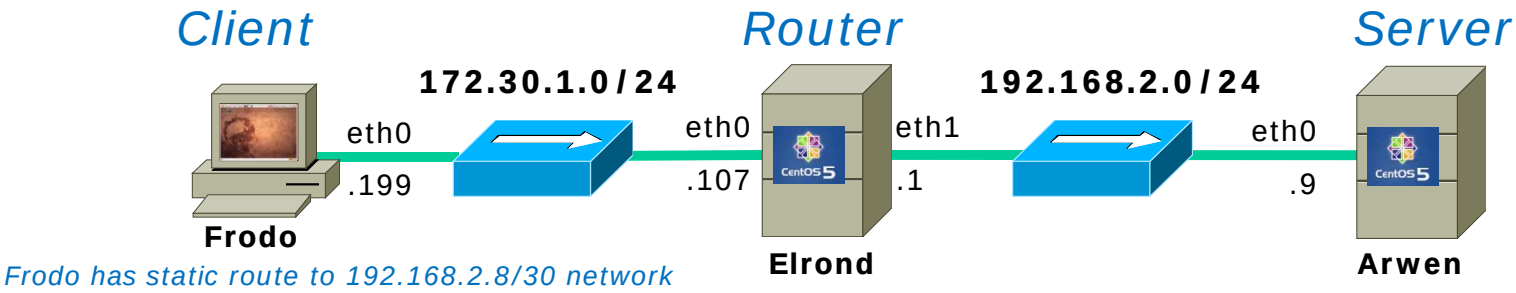
Table: filter
Chain: FORWARD



No Rules

Chain Policy: ACCEPT

Netfilter – examples



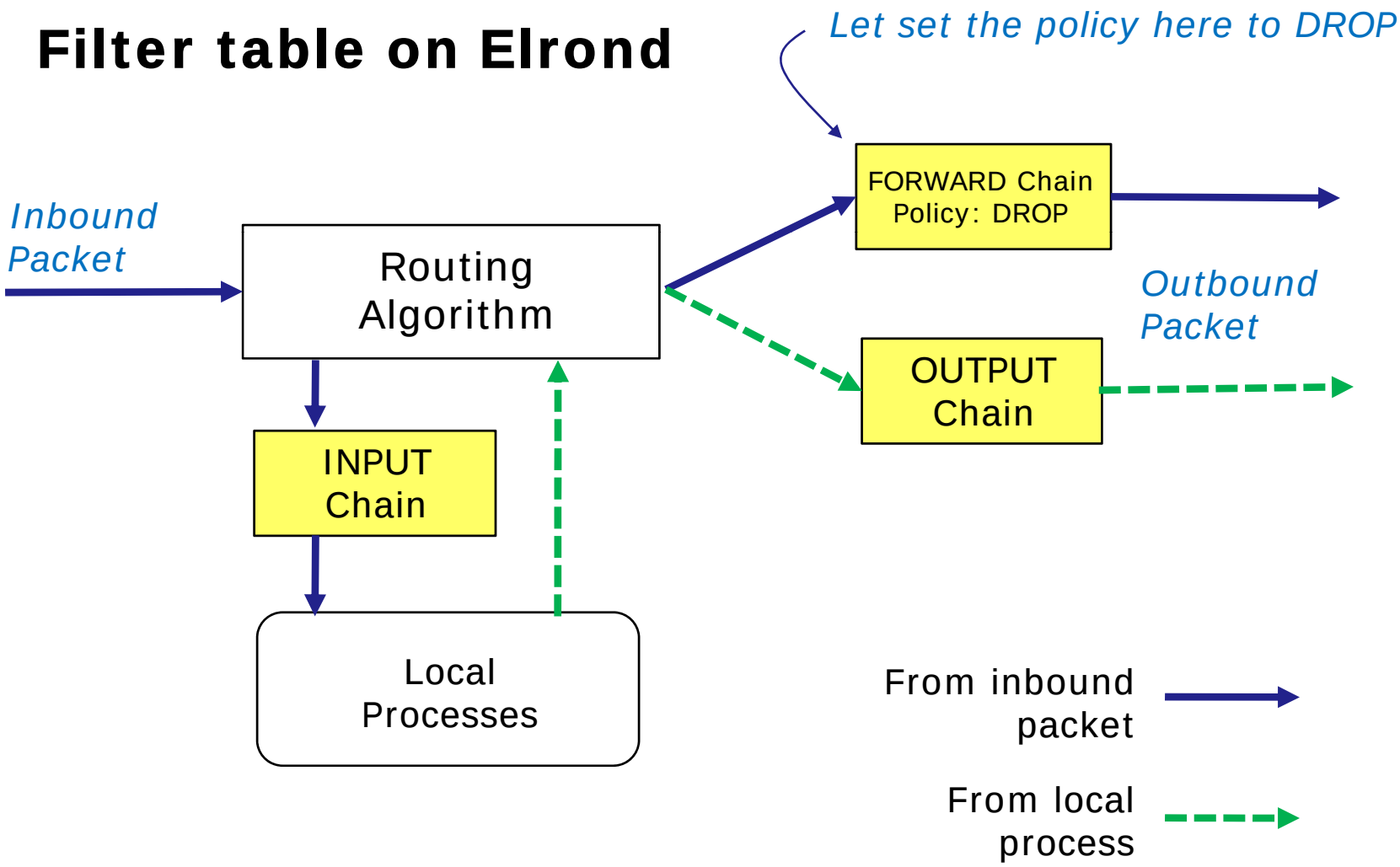
Frodo has static route to 192.168.2.8/30 network

```
[root@elrond ~]# iptables -P FORWARD ACCEPT
[root@elrond ~]# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@elrond ~]#
```

Frodo can ping via Elrond to Arwen because Elrond's FORWARD chain's policy is ACCEPT

```
root@frodo:~# ping -c 2 arwen
PING arwen (192.168.2.9) 56(84) bytes of data.
64 bytes from arwen (192.168.2.9): icmp_seq=1 ttl=63 time=5.38 ms
64 bytes from arwen (192.168.2.9): icmp_seq=2 ttl=63 time=1.13 ms

--- arwen ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1005ms
```

Netfilter – examples



Netfilter – examples

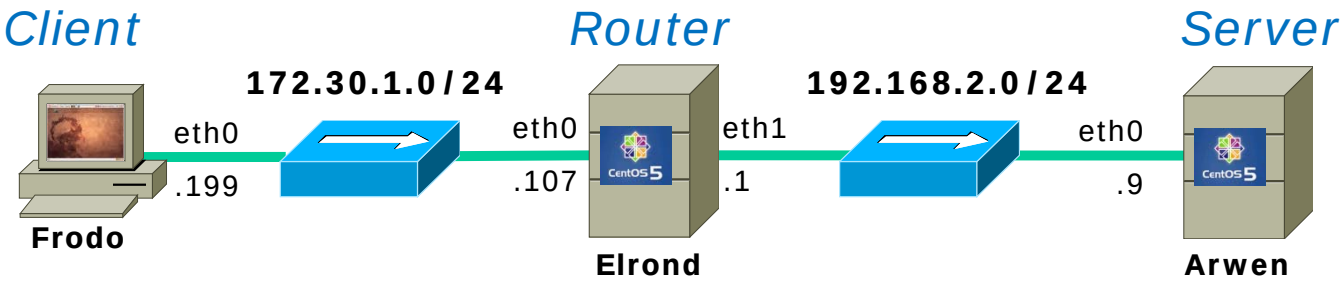
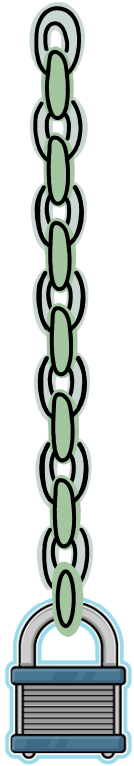


Table: filter
Chain: FORWARD

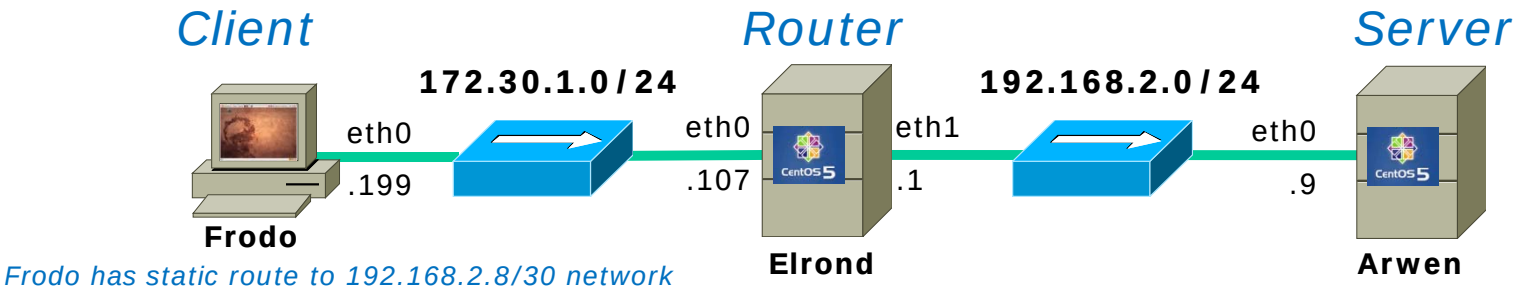


No Rules

Chain Policy: DROP

DROP everything else

Netfilter – examples



Frodo has static route to 192.168.2.8/30 network

```
[root@elrond ~]# iptables -P FORWARD DROP
[root@elrond ~]# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@elrond ~]#
```

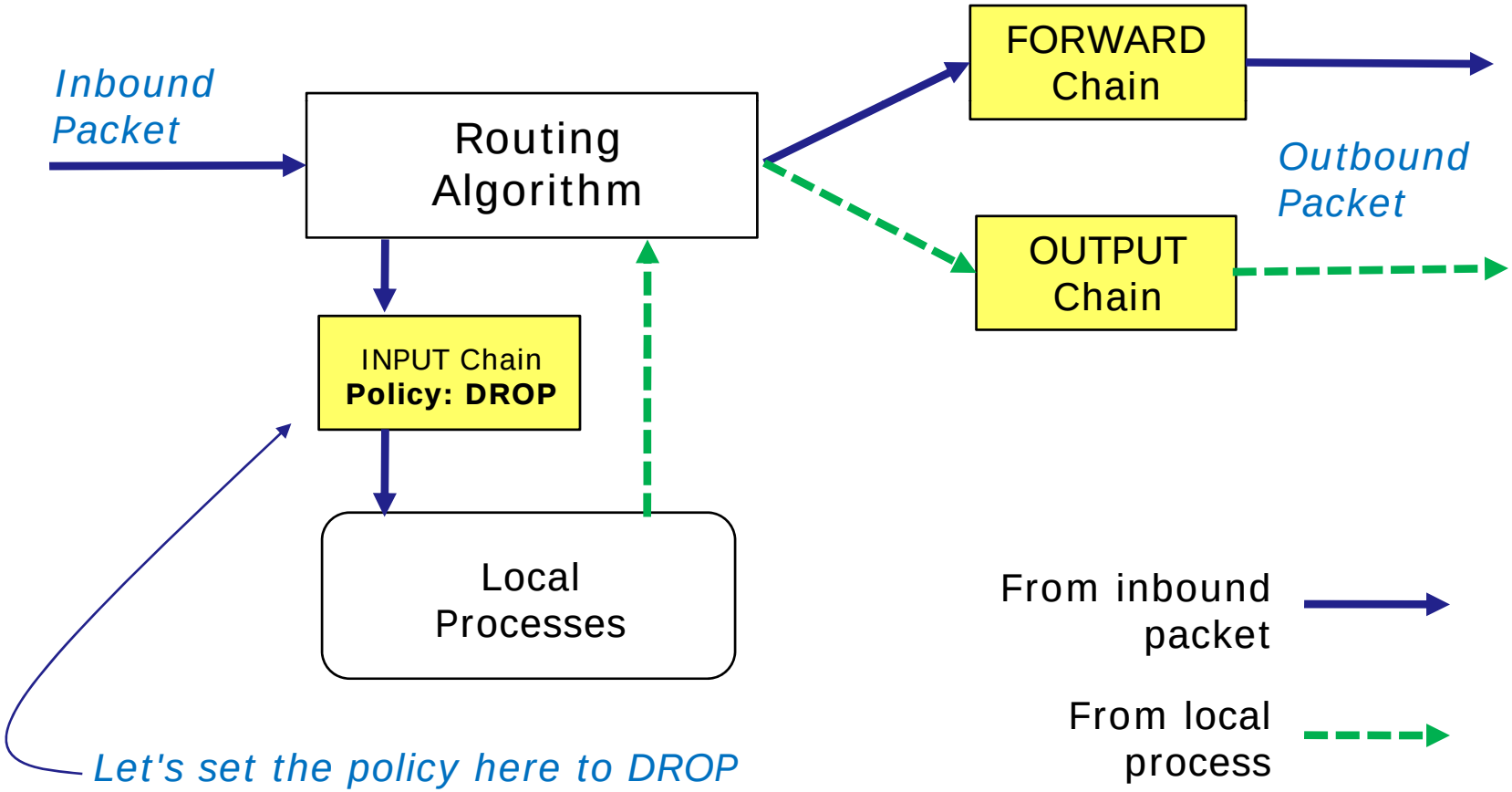
Frodo cannot ping Arwen via Elrond because Elrond's FORWARD chain policy is DROP

```
root@frodo:~# ping -c 2 arwen
PING arwen (192.168.2.9) 56(84) bytes of data.

--- arwen ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1004ms
```

Table: filter
Chain: INPUT
IP address rules

Netfilter – examples

Filter table on Elrond



Netfilter – examples

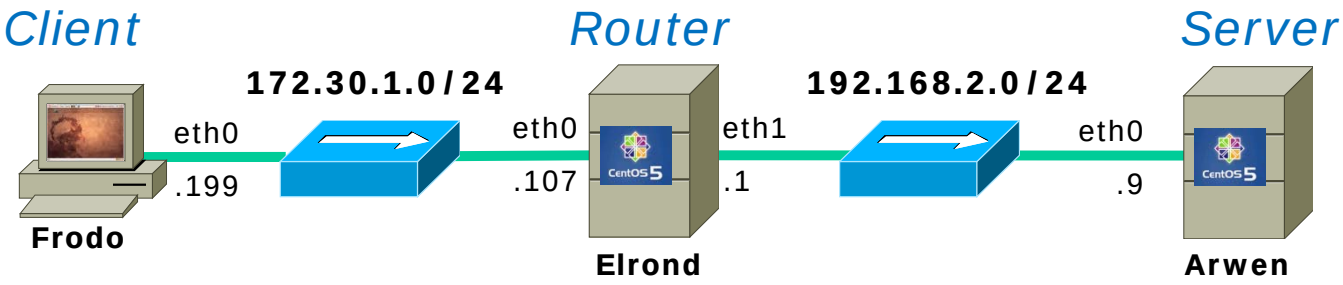
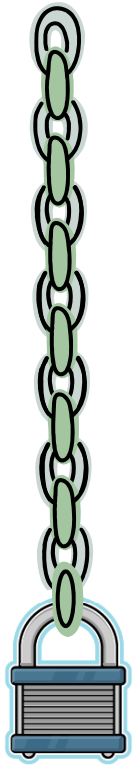


Table: filter
Chain: INPUT



Chain Rules:

-s 172.30.1.199/32 -j REJECT

Reject anything from Frodo

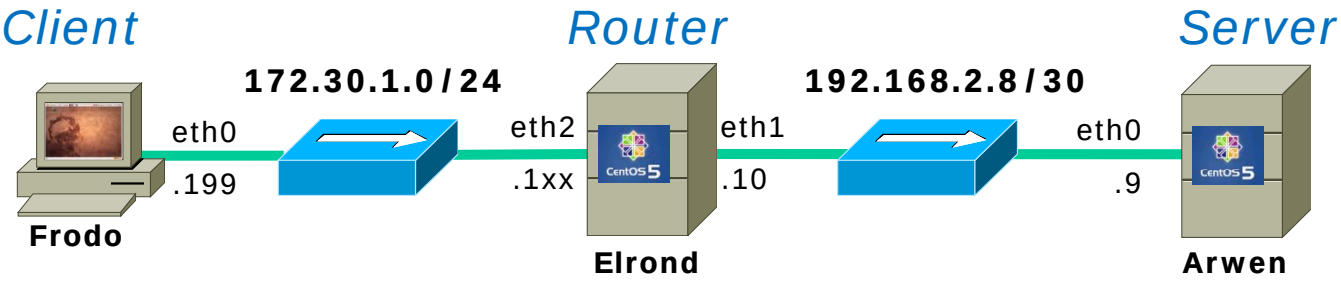
-s 192.168.0.0/16 –j ACCEPT

Accept all packets from 192.168.x.x

Chain Policy: DROP

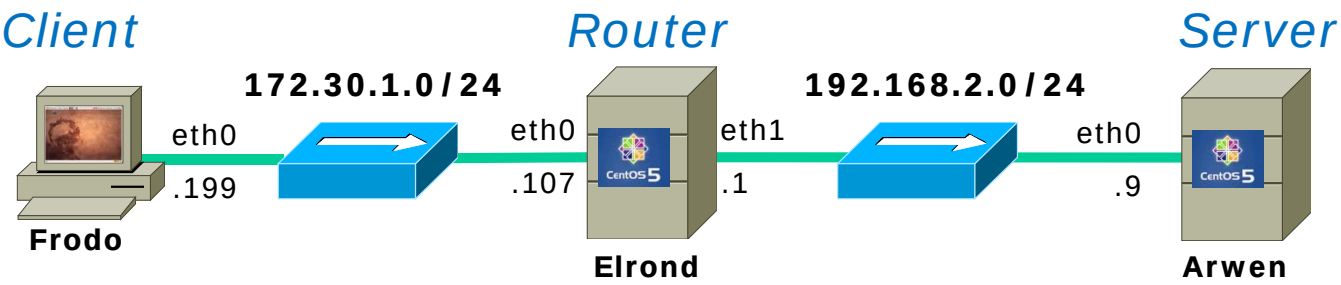
DROP everything else

Netfilter – examples




```
[root@elrond ~]# iptables -F
[root@elrond ~]# iptables -A INPUT -s 172.30.1.199/32 -j REJECT
[root@elrond ~]# iptables -A INPUT -s 192.168.0.0/16 -j ACCEPT
[root@elrond ~]# iptables -L -n
Chain INPUT (policy DROP)
target     prot opt source               destination
REJECT     all  --  172.30.1.199         0.0.0.0/0           reject-with
icmp-port-unreachable
ACCEPT     all  --  192.168.0.0/16       0.0.0.0/0

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@elrond ~]#
```

Netfilter – examples

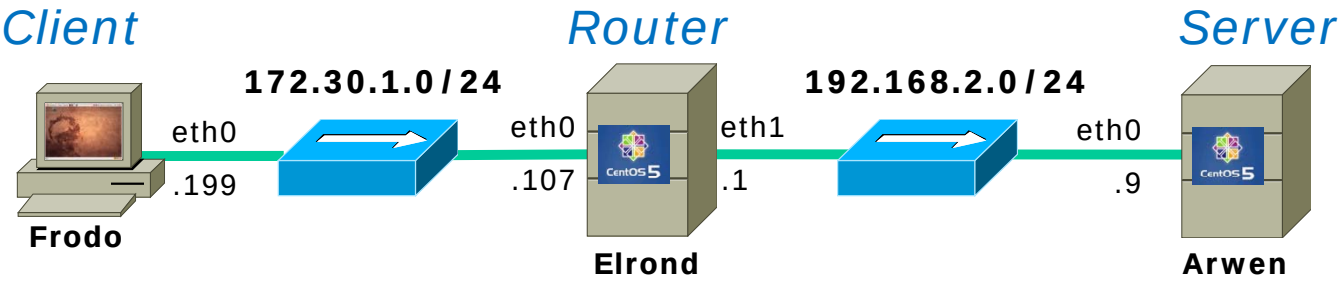



```
Chain INPUT (policy DROP)
target     prot opt source               destination
REJECT     all  --  172.30.1.199         0.0.0.0/0           reject-with
icmp-port-unreachable
ACCEPT     all  --  192.168.0.0/16       0.0.0.0/0
```

```
root@frodo:~# ping -c 2 elrond
PING elrond (172.30.1.107) 56(84) bytes of data.
From elrond (172.30.1.107) icmp_seq=1 Destination Port Unreachable
From elrond (172.30.1.107) icmp_seq=2 Destination Port Unreachable

--- elrond ping statistics ---
2 packets transmitted, 0 received, +2 errors, 100% packet loss, time 1004ms

root@frodo:~#
```

Ping from Frodo to Elrond fails with port unreachable

Netfilter – examples

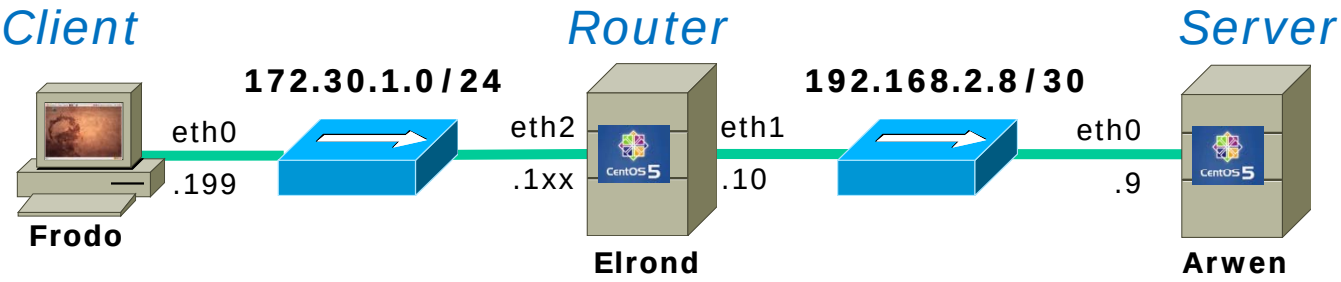



```
Chain INPUT (policy DROP)
target     prot opt source               destination
REJECT     all  --  172.30.1.199         0.0.0.0/0           reject-with
icmp-port-unreachable
ACCEPT     all  --  192.168.0.0/16       0.0.0.0/0
```

```
[root@arwen ~]# ping -c2 elrond
PING elrond (192.168.2.10) 56(84) bytes of data.
64 bytes from elrond (192.168.2.10): icmp_seq=1 ttl=64 time=5.86 ms
64 bytes from elrond (192.168.2.10): icmp_seq=2 ttl=64 time=1.74 ms

--- elrond ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 1.748/3.807/5.867/2.060 ms
[root@arwen ~]#
```

Ping from Arwen to Elrond succeeds

Netfilter – examples




```
Chain INPUT (policy DROP)
target     prot opt source               destination
REJECT     all  --  172.30.1.199         0.0.0.0/0           reject-with
icmp-port-unreachable
ACCEPT     all  --  192.168.0.0/16       0.0.0.0/0
```

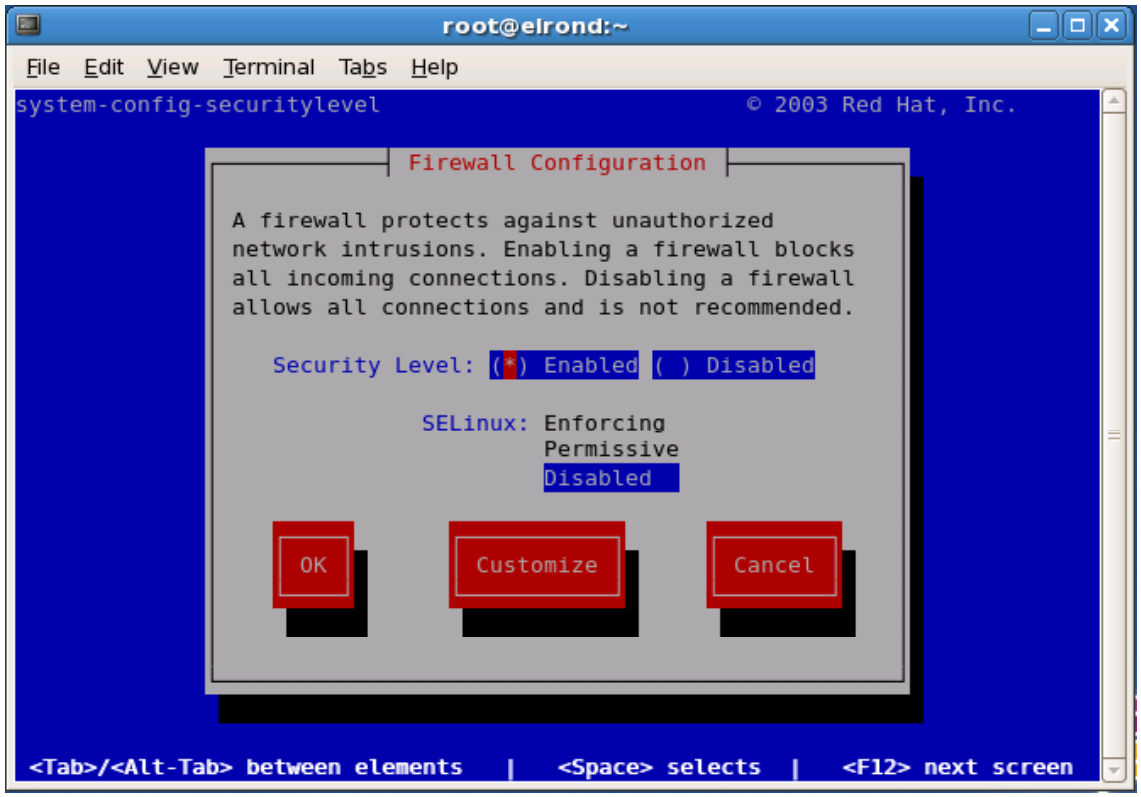
```
[root@nosmo root]# ping -c 2 elrond
PING elrond (172.30.1.107) 56(84) bytes of data.

--- elrond ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1012ms

[root@nosmo root]#
```

Ping from Nosmo (172.30.1.1) to Elrond fails, timing out without any error messages

Lab 5

lokkit

```
[root@elrond ~]# lokkit
```



Settings kept in: /etc/sysconfig/system-config-securitylevel

Lokkit command for enabling and disabling firewall and SELinux settings.

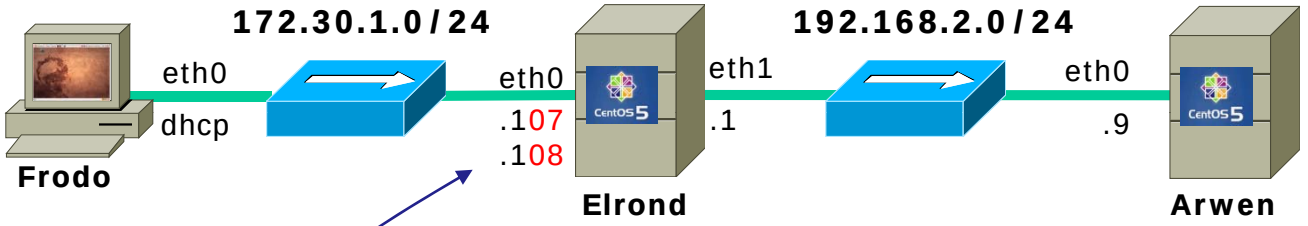
*Beware, this tool can overwrite any firewall settings you have made with **iptables** commands*

Frodo is an outside host that is allowed to use the Telnet Server on Arwen

Shire
(Outside)

Rivendell
(Inside)

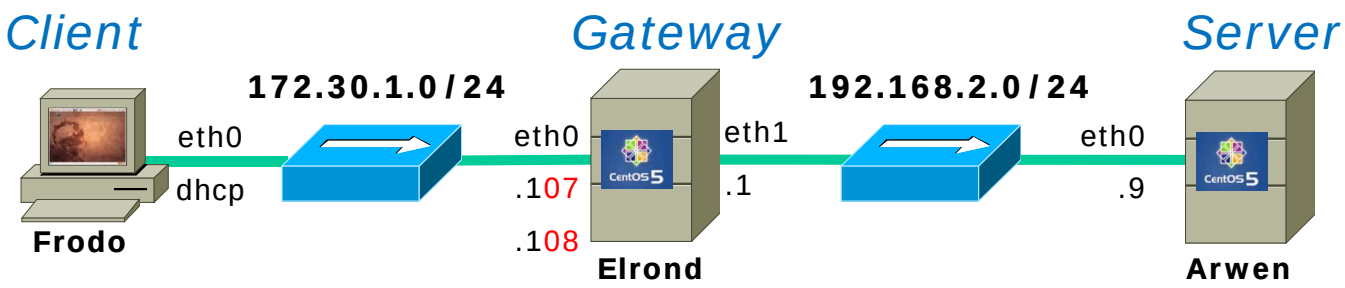
Telnet server is installed on Arwen



These address are for example only. Only use the static IP addresses designated for you station!

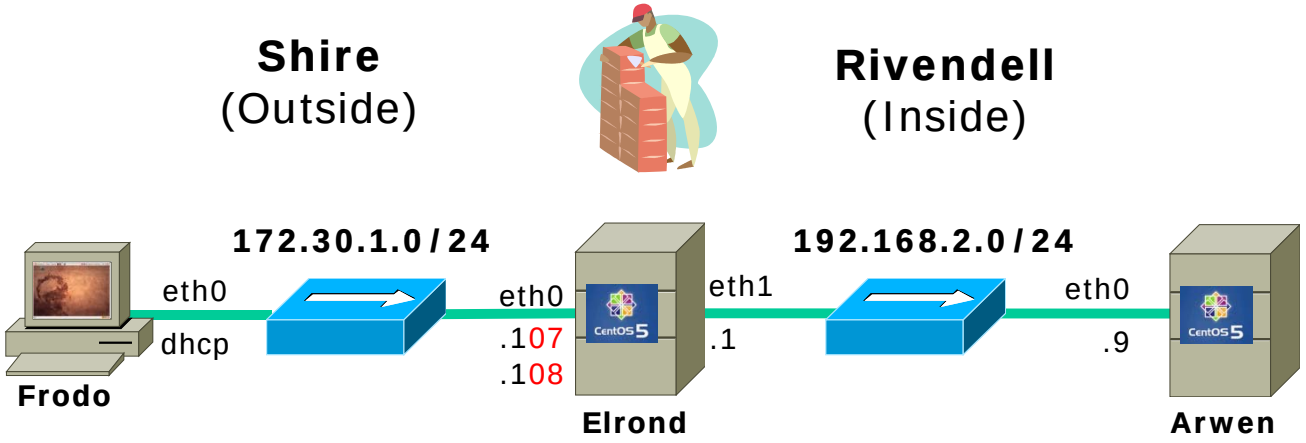
We create a firewall on Elrond, the gateway

NAT is also configured here so Rivendell hosts have Internet access



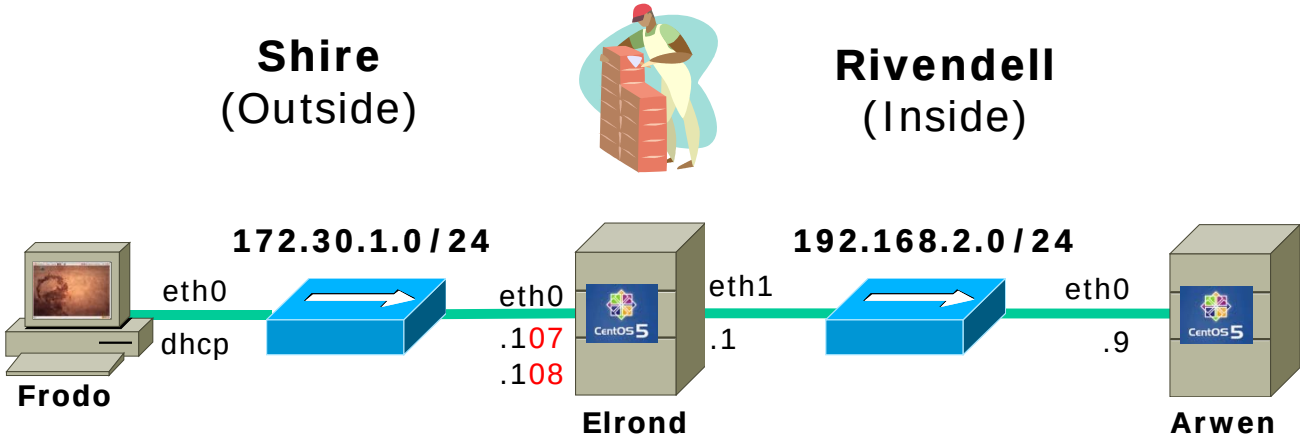

```
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP

iptables -A FORWARD -s 192.168.2.0/24 -d 0/0 -m state --state NEW -j ACCEPT
iptables -A FORWARD -s 0/0 -d 192.168.2.9 -m state --state NEW,ESTABLISHED,RELATED -p tcp --dport 23 -j ACCEPT
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

iptables -A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

iptables -A INPUT -i eth1 -s 192.168.2.0/24 -d 192.168.2.1 -m state --state NEW -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT


ifconfig eth0:1 172.30.1.108 netmask 255.255.255.0 broadcast 172.30.1.255

iptables -t nat -A PREROUTING -i eth0 -d 172.30.1.108 -j DNAT --to-destination 192.168.2.9
iptables -t nat -A POSTROUTING -o eth0 -s 192.168.2.9 -j SNAT --to-source 172.30.1.108
iptables -t nat -A POSTROUTING -o eth0 -s 192.168.2.0/24 -j SNAT --to-source 172.30.1.107

iptables -A INPUT -j LOG --log-level info --log-prefix "iptables INPUT: "
iptables -A FORWARD -j LOG --log-level info --log-prefix "iptables FORWARD: "
```

Note: Your Elrond static IP will be based on the station you use



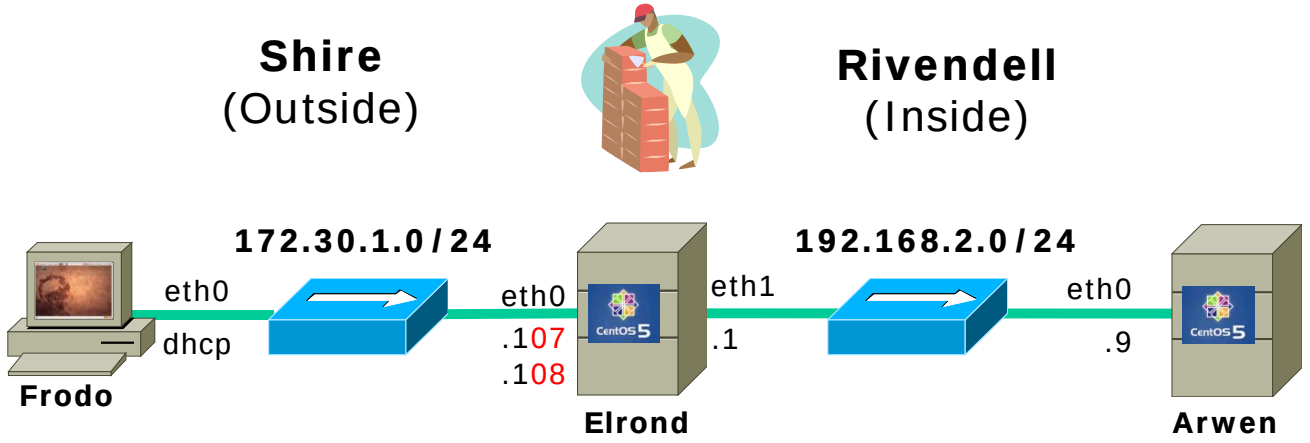

```
root@frodo:~# telnet 172.30.1.108
Trying 172.30.1.108...
Connected to 172.30.1.108.
Escape character is '^]'.
CentOS release 5.2 (Final)
Kernel 2.6.18-92.1.22.el5 on an i686
login: cis192
Password:
Last login: Mon Mar 16 23:13:09 from 172.30.1.195
[cis192@arwen ~]$ exit
```

From Frodo we have access to the Telnet server on Arwen using the public IP address on Elrond




```
[root@arwen ~]# ping google.com
PING google.com (74.125.67.100) 56(84) bytes of data.
64 bytes from gw-in-f100.google.com (74.125.67.100): icmp_seq=1 ttl=243 time=221 ms
64 bytes from gw-in-f100.google.com (74.125.67.100): icmp_seq=2 ttl=243 time=204 ms

--- google.com ping statistics ---
3 packets transmitted, 2 received, 33% packet loss, time 2003ms
rtt min/avg/max/mdev = 204.217/212.833/221.450/8.628 ms
[root@arwen ~]#
```

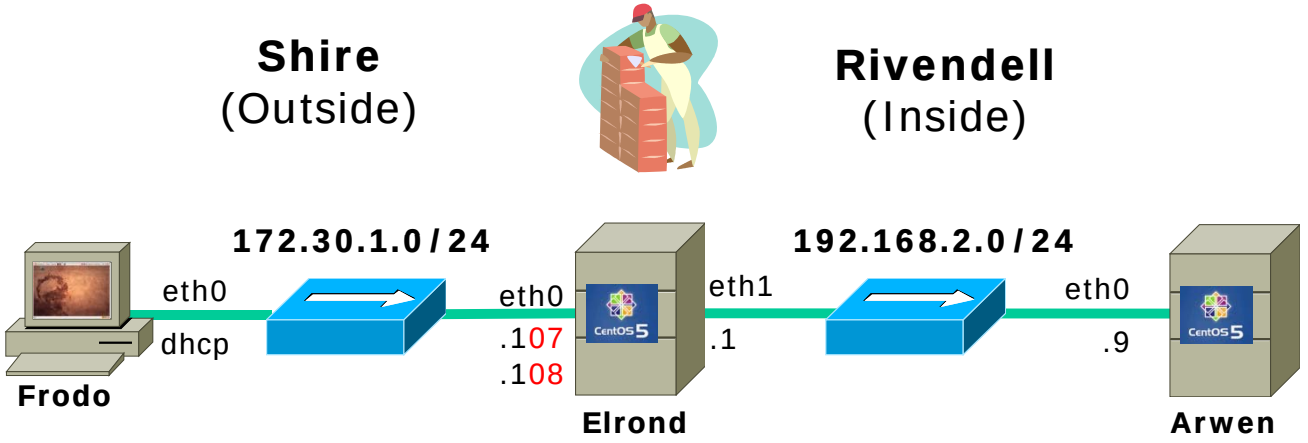
Arwen, which is on a private network, has Internet access via NAT on Elrond. No static routes needed!




```
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP
```

Set the policies on each filter chain to DROP. If no rules in the chains match then the packets will be dropped.

DROP is "silent" - no error messages in a response are sent back



iptables -A FORWARD -s 192.168.2.0/24 -d 0/0 -m state --state NEW -j ACCEPT

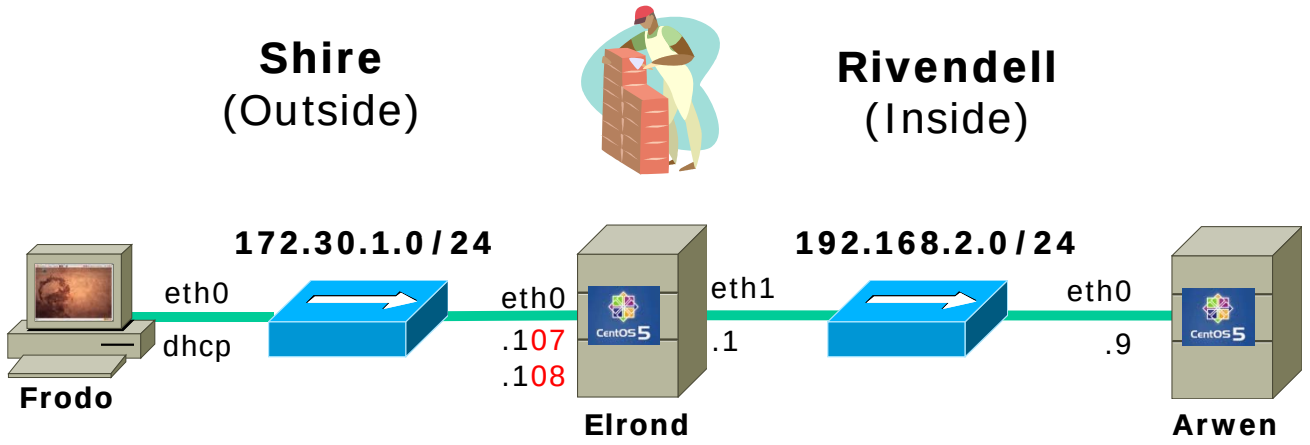
All new packets from Rivendell hosts (Arwen) will be forwarded

iptables -A FORWARD -s 0/0 -d 192.168.2.9 -m state --state NEW,ESTABLISHED,RELATED -p tcp --dport 23 -j ACCEPT

All Telnet packets going to the Telnet Server will be forwarded

iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

All related and established connection packets will be forwarded

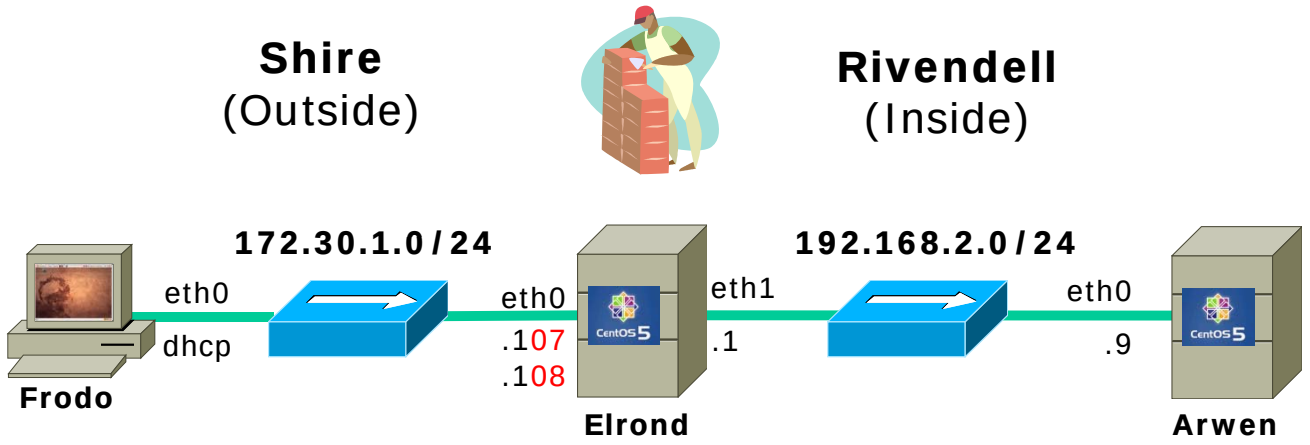


iptables -A OUTPUT -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

All packets from Elrond are allowed out

iptables -A INPUT -i eth1 -s 192.168.2.0/24 -d 192.168.2.1 -m state --state NEW -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

Allow any new incoming connections as well as ongoing traffic from Rivendell hosts

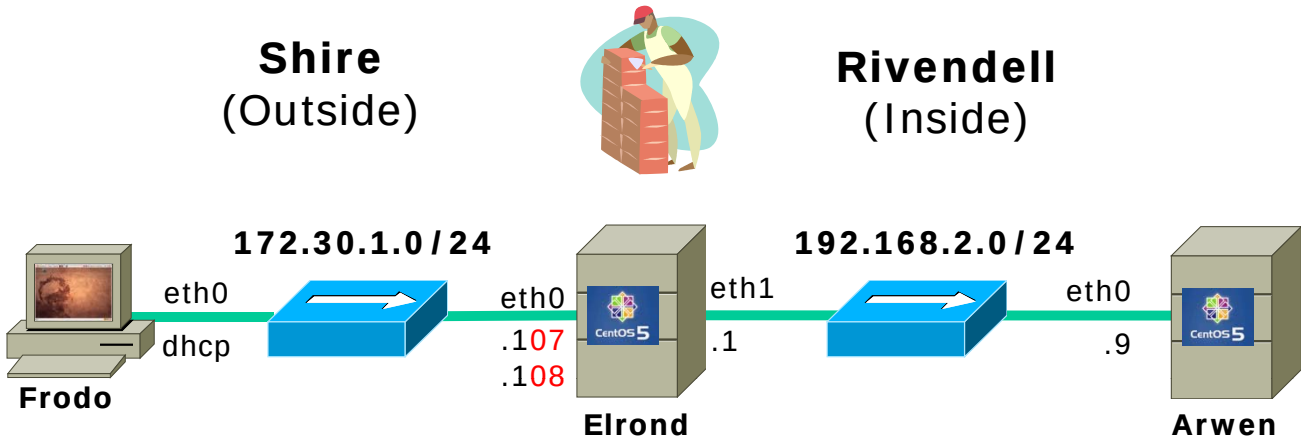


iptables -t nat -A PREROUTING -i eth0 -d 172.30.1.108 -j DNAT --to-destination 192.168.2.9

Translate any incoming packets to the public IP address to Arwen

iptables -t nat -A POSTROUTING -o eth0 -s 192.168.2.9 -j SNAT --to-source 172.30.1.108
iptables -t nat -A POSTROUTING -o eth0 -s 192.168.2.0/24 -j SNAT --to-source 172.30.1.107

NAT outgoing Arwen packets to use the second public IP address (used for the Telnet server), for the other Rivendell hosts we will NAT to Elrond's first public IP address.




```
iptables -A INPUT -j LOG --log-level info --log-prefix "iptables INPUT: "
iptables -A FORWARD -j LOG --log-level info --log-prefix "iptables FORWARD: "
```

Log firewall events for the INPUT and FORWARD chains

Wrap

New commands, daemons and files:

- iptables
- netstat
- service
- yum

Daemons and related configuration files

tcpd	/etc/hosts.allow,hosts.deny

Next Class

Assignment: Check Calendar Page
http://simms-teach.com/cis192calendar.php

Lab 5 due

Quiz questions for next class:

- How do you display the current filter table chains?
- How do you display the current nat table chains?
- How do set the FORWARD chain policy to ACCEPT?

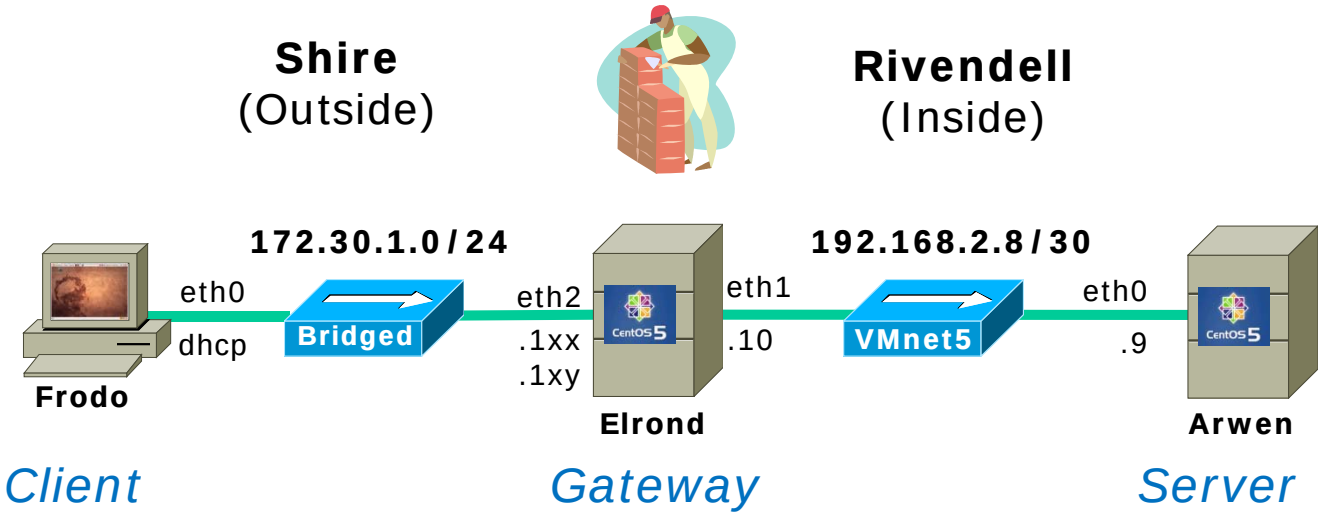
Backup

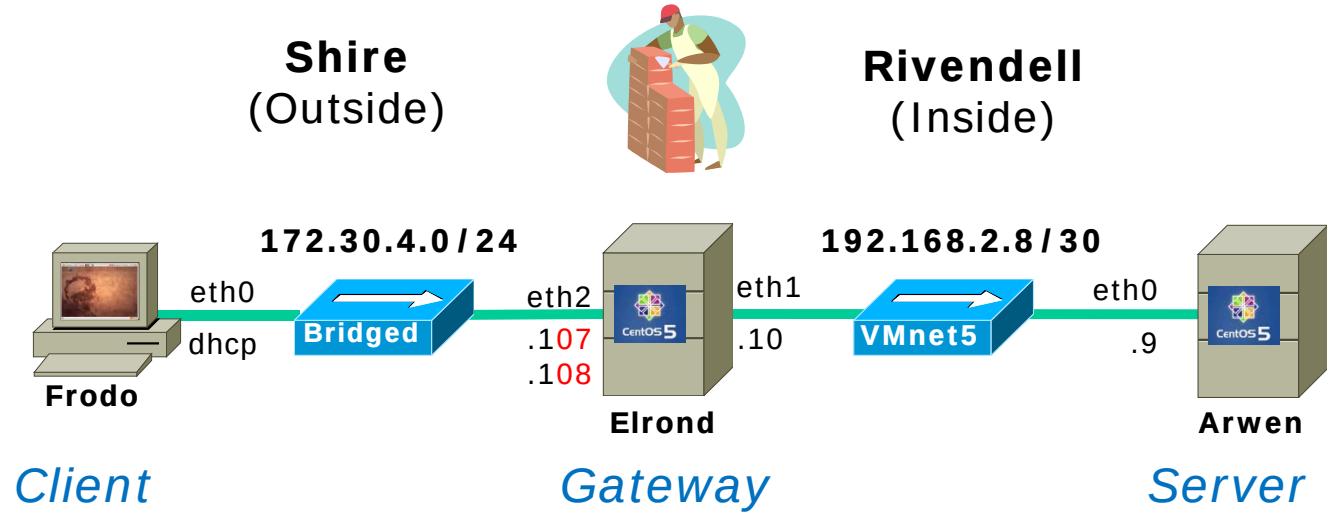
IP addresses for VM's in the classroom

Station	IP	Static 1
Instructor	172.30.1.100	172.30.1.125
Station-01	172.30.1.101	172.30.1.126
Station-02	172.30.1.102	172.30.1.127
Station-03	172.30.1.103	172.30.1.128
Station-04	172.30.1.104	172.30.1.129
Station-05	172.30.1.105	172.30.1.130
Station-06	172.30.1.106	172.30.1.131
Station-07	172.30.1.107	172.30.1.132
Station-08	172.30.1.108	172.30.1.133
Station-09	172.30.1.109	172.30.1.134
Station-10	172.30.1.110	172.30.1.135
Station-11	172.30.1.111	172.30.1.136
Station-12	172.30.1.112	172.30.1.137

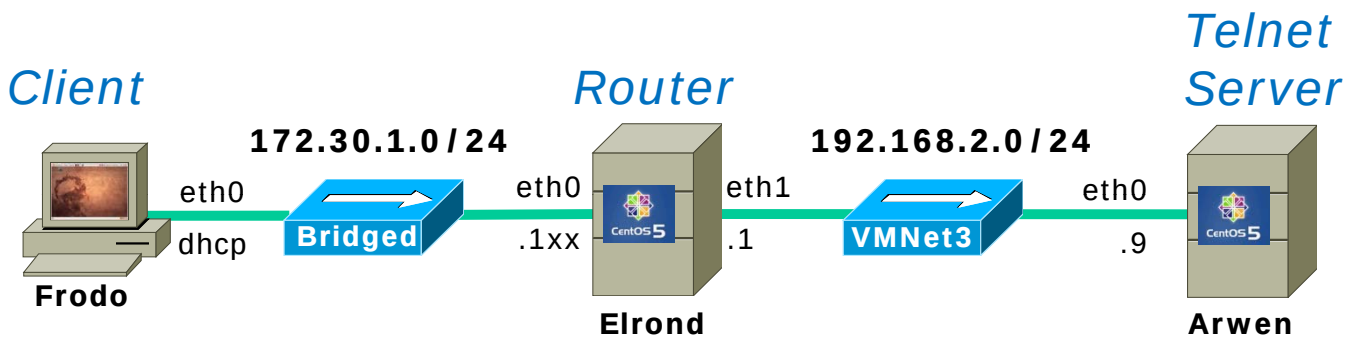
Station	IP	Static 1
Station-13	172.30.1.113	172.30.1.138
Station-14	172.30.1.114	172.30.1.139
Station-15	172.30.1.115	172.30.1.140
Station-16	172.30.1.116	172.30.1.141
Station-17	172.30.1.117	172.30.1.142
Station-18	172.30.1.118	172.30.1.143
Station-19	172.30.1.119	172.30.1.144
Station-20	172.30.1.120	172.30.1.145
Station-21	172.30.1.121	172.30.1.146
Station-22	172.30.1.122	172.30.1.147
Station-23	172.30.1.123	172.30.1.148
Station-24	172.30.1.124	172.30.1.149

Note the static IP address for your station to use in the next class exercise

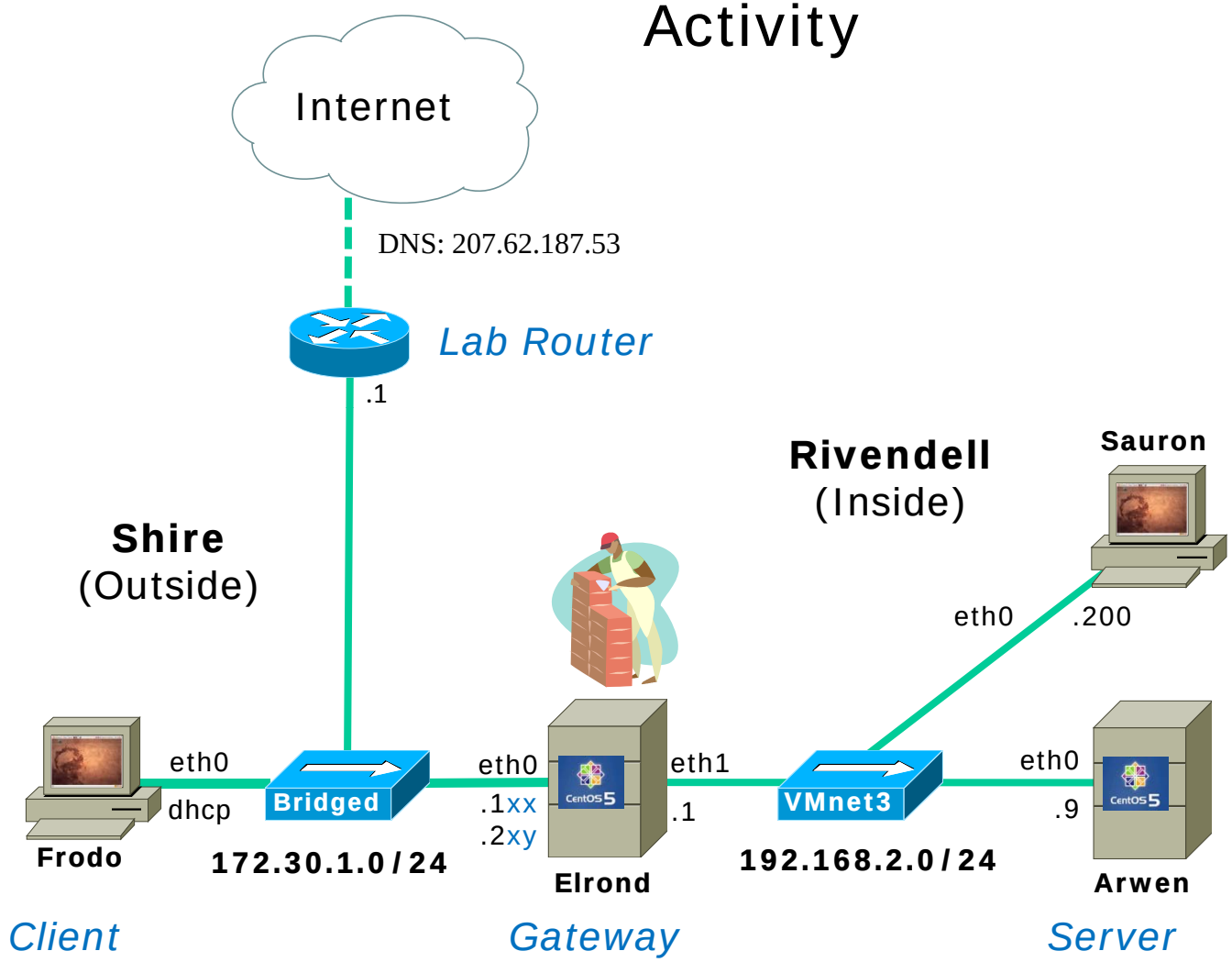

Shire
(Outside)
Rivendell
(Inside)
172.30.1.0/24
192.168.2.8/30
eth0
dhcp
Bridged
eth2
.1xx
.1xy
eth1
.10
VMnet5
eth0
.9
Frodo
Elrond
Arwen
Client
Gateway
Server


Shire
(Outside)
Rivendell
(Inside)
172.30.4.0/24
192.168.2.8/30
eth0
dhcp
Bridged
eth2
.107
.108
eth1
.10
VMnet5
eth0
.9
Frodo
Elrond
Arwen
Client
Gateway
Server

Exercise


Client
Router
Telnet Server
172.30.1.0/24
192.168.2.0/24
eth0
dhcp
Bridged
eth0
.1xx
eth1
.1
VMNet3
eth0
.9
Frodo
Elrond
Arwen

Activity



- Build this
- No static routes

Installing and Configuring Telnet

Note: there may be multiple telnetd daemons

```
[root@arwen ~]# ls /etc/xinetd.d/*tel*
/etc/xinetd.d/ekrb5-telnet  /etc/xinetd.d/telnet
/etc/xinetd.d/krb5-telnet   /etc/xinetd.d/telnet.rpmsave

[root@arwen ~]# cat /etc/xinetd.d/krb5-telnet
# default: off
# description: The kerberized telnet server accepts normal telnet sessions, \
#              but can also use Kerberos 5 authentication.
service telnet
{
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/kerberos/sbin/telnetd
        log_on_failure  += USERID
        disable         = yes
}
[root@arwen ~]#
```

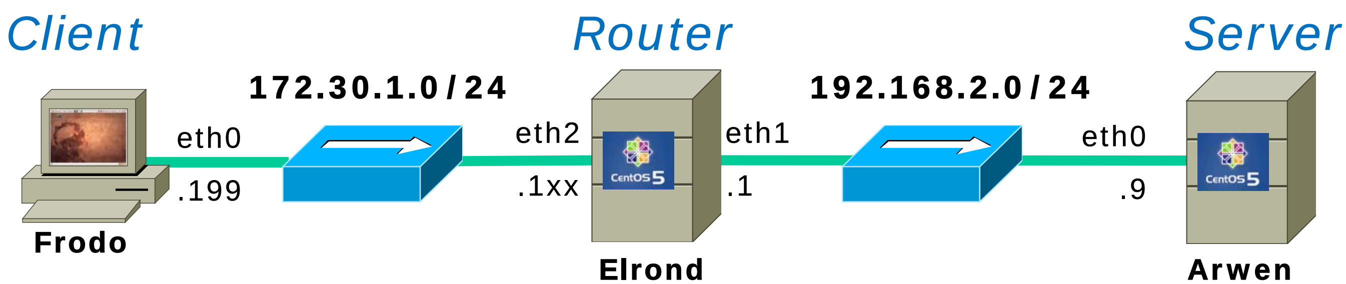
Disable this telnet daemon

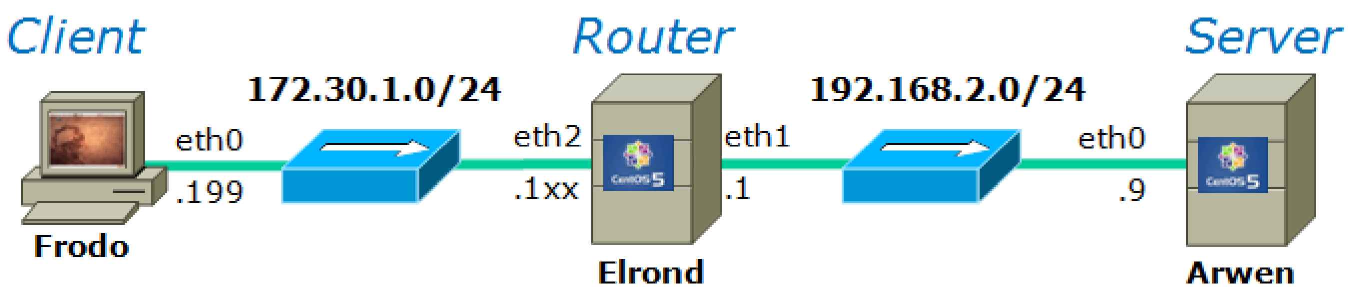
Installing and Configuring Telnet

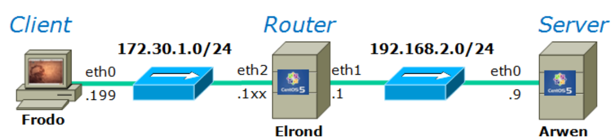
Note: there may be multiple telnetd daemons

```
[root@arwen ~]# ls /etc/xinetd.d/*tel*
/etc/xinetd.d/ekrb5-telnet  /etc/xinetd.d/telnet
/etc/xinetd.d/krb5-telnet   /etc/xinetd.d/telnet.rpmsave

[root@arwen ~]# cat /etc/xinetd.d/ekrb5-telnet
# default: off
# description: The kerberized telnet server accepts only telnet sessions, \
#              which use Kerberos 5 authentication and encryption.
service telnet
{
        flags           = REUSE
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/kerberos/sbin/telnetd
        server_args     = -e
        log_on_failure  += USERID
        disable         = yes
}
[root@arwen ~]#
```

Disable this telnet daemon


Client
Router
Server
172.30.1.0/24
192.168.2.0/24
eth0
.199
Frodo
eth2
.1xx
eth1
.1
Elrond
eth0
.9
Arwen


Client
Router
Server
172.30.1.0/24
192.168.2.0/24
eth0
.199
Frodo
eth2
.1xx
eth1
.1
Elrond
eth0
.9
Arwen


Client
Router
Server
172.30.1.0/24
192.168.2.0/24
eth0
.199
Frodo
eth2
.1xx
eth1
.1
Elrond
eth0
.9
Arwen