

# Ateneo Cervini-Eliazo Networks

## Linux Networking



Ateneo Cervini-Eliazo Networks

30 January 2001

<http://cersa.admu.edu.ph/>

[william.s.yu@ieee.org](mailto:william.s.yu@ieee.org)

## Section I

# Introduction

## Components

- ★ Hardware
- ★ Configuration
- ★ Firewalling
- ★ Masquerading
- ★ Other Security Items

## Section II

# Hardware

## Classes of Linux Networking Devices

- ★ lo - the local loopback interface
- ★ ethX - Ethernet device interfaces
- ★ trX - Token Ring device interfaces
- ★ slX - SLIP interfaces
- ★ pppX - PPP interfaces
- ★ plipX - PLIP interfaces
- ★ axX - AX.25 interfaces (ham radio)

## Installation and Configuration

- ★ ensure that the appropriate kernel driver for your network device is included in the kernel or compiled as a module
- ★ recompile kernel or
- ★ call **modprobe** to load the appropriate kernel module.  
eg. `modprobe 3c509`
- ★ in some instances the NICs cannot be autoprobeed. Insert the following line into the `lilo.conf`:  
`append="ether=irq,base_addr,name"`  
For Example:  
`append="ether=10,300,eth0"`

## Section III

# Configuration

## Steps for Network Configuration

- ★ collect all necessary networking information
- ★ install network devices
- ★ load interface
- ★ configure necessary routes
- ★ check if interfaces are working

## The **BIG FOUR** Linux Networking Commands

- ★ ifconfig - configure network interfaces
- ★ route - configure network routes
- ★ netstat - monitoring connections
- ★ arp - monitoring address tables

## Configuring Network Interfaces

```
ifconfig [interface] [ip] broadcast [broadcast] netmask [netmask]
```

- ★ configure kernel level interfaces
- ★ sets the IP address, network mask, broadcast address and network address of this interface.
- ★ Example: `ifconfig eth0 192.168.0.1 broadcast 192.168.255.255 netmask 255.255.0.0`

## Configuring Network Routes

route [operation] [network address] netmask [netmask] [interface] metric [metric number]

- ★ sets up the static routes to be used in the network
- ★ Example: route add -net 192.168.0.0 netmask 255.255.0.0 eth0 metric 1
- ★ Example: route add default gw 192.168.0.1 netmask 255.255.0.0 metric 1

## Monitoring Connections

```
netstat [-rvenaoc] [-tcp--t] [-udp--u] [-raw--w] [-groups--g] [-unix--x]  
[-inet--ip] [-ax25] [-ipx] [-netrom]
```

- ★ display network connections, routing tables, interface statistics, masquerade connections, netlink messages, and multicast memberships
- ★ Example: netstat
- ★ Example: netstat -rn

## Monitoring Address Tables

`arp [-dv] [hostname]`

★ manipulate system ARP cache, MAC address mappings

★ Example: `arp`

★ Example: `arp -d 192.168.0.1`

## Check If Network is Working

- ★ ping - sends ICMP packets to determine if host is available
- ★ traceroute - print the route packets take to network host

## Section IV

# Firewall

## Basics and the Need

- ★ is a secure and trusted machine that sits between a private network and a public network
- ★ is configured with a set of rules that determine which network traffic will be allowed to pass and which will be blocked or refused
- ★ sometimes located inside their corporate network to segregate sensitive areas of the organization from other employees

## Types of Firewalls

There are typically two classes of firewalls:

- ★ Application-level firewall
- ★ Network-level firewall (Packet Filter)

There are also two classes of packet-level firewalls:

- ★ Stateful
- ★ Stateless

## Application-level Firewalls

- ★ Proxy (squid)
- ★ Socks (dante, NEC socks)
- ★ HTTP Acceleration (squid, apache)
- ★ TIS Firewall Toolkit <<http://www.fwtk.org/>>

## Network-level Firewalls

- ★ ipfwadm (Linux Kernel 2.0.x) - deprecated
- ★ ipchains (Linux Kernel 2.2.x)
- ★ iptables/netfilter (Linux Kernel 2.4.x)

## Network-level Firewalls

- ★ Source/Destination Address
- ★ Protocol
- ★ Port Number
- ★ Content (only for netfilter)

## Kernel Configuration

For 2.2 Kernels:

- ★ Network firewalls
- ★ TCP/IP networking
- ★ IP: firewalling
- ★ IP: firewall packet logging

For 2.4 Kernels:

- ★ Network packet filtering (replaces ipchains)

## Configuring Firewall Rules

```
ipchains -[operation] [type] -s [source] -d [destination] -j [rule]
```

- ★ sets up a chain by performing the operation on type that is qualified by the source and destination address and performs the rule on them
- ★ Example: `ipchains -A forward -s 192.168.0.1 -j MASQ`
- ★ Example: `ipchains -A forward -s 0/0 -d 0/0 6000:8999 -j DENY`

## **Section V**

# Masquerading

## Basics and the Need

- ★ also known as Network Address Translation
- ★ address the shortage of routable IPs
- ★ allows you to use a private (reserved) IP network address on your LAN and have your Linux-based router perform some clever, real-time translation of IP addresses and ports
- ★ a masquerading server pretends to be all the hosts in the internal network

## Kernel Configuration

For 2.2 Kernels:

- ★ Network firewalls
- ★ TCP/IP networking
- ★ IP: firewalling
- ★ IP: masquerading
- ★ IP: firewall packet logging
- ★ IP: ipautofw masq support
- ★ IP: ICMP masquerading

For 2.4 Kernels:

- ★ Network packet filtering (replaces ipchains)

## Configuring Masquerading

Disable all forwarding:

```
ipchains -P forward -j deny
```

Masquerade all hosts in the subnet 192.168.1.0/24

```
ipchains -A forward -s 192.168.1.0/24 -d 0/0 -j  
MASQ
```

## Section VI

# Other Security Items

## Inetd or Xinetd Superserver

- ★ /etc/inetd.conf for inetd
- ★ /etc/inetd.d/ for Xinetd
- ★ daemon that starts Internet services
- ★ only daemon that listens to all service ports
- ★ enables the use of tcpwrappers
- ★ generally disable services not in use
- ★ replace unsecure services with secure versions

## **/etc/services**

- ★ list of Internet services
- ★ maps a service with its corresponding ports
- ★ list must contain all of your services' ports

## **tcpwrappers**

- ★ defined in two files `/etc/hosts.allow` and `/etc/hosts.deny`
- ★ also known as host access files
- ★ limits access to services defined by the `inetd` or `xinetd` daemon

## Access Control Language

- ★ `/etc/hosts.allow` - refers to hosts which are allowed to access certain services
- ★ `/etc/hosts.deny` - refers to hosts which are not allowed to access certain services
- ★ syntax: `daemon:clients:[commands]`
- ★ example: `ALL:.pvao.gov.ph`
- ★ example: `in.fingerd:.pvao.gov.ph EXCEPT veterans.pvao.gov.ph`
- ★ example:  

```
in.tftpd: ALL: (/some/where/safe_finger -l @%h |  
/usr/ucb/mail -s %d-%h root) &
```

## Disable Spoofing

- ★ Linux uses a resolver library to obtain the IP address corresponding to a hostname
- ★ enabling IP spoof is a security measure that prevents hosts from pretending to be a different host

Sample /etc/resolv.conf file

```
order bind,hosts
multi on
nospoof on
```

## ***/etc/securetty***

- ★ terminals that root is allowed to login from
- ★ disable all pseudo-terminal to prevent root from logging remotely

Sample */etc/securetty* file

```
tty1
```

```
tty2
```

## Section XI

# Conclusion



Copyright © 2000-2001 by William Emmanuel S. Yu. This material may be distributed only subject to the terms and conditions set forth in the Open Content License, v1.0 or later (the latest version is presently available at <http://opencontent.org/opl.shtml>).