TCP/IP Networking and Linux

By David F. Skoll

Roaring Penguin Software Inc.

17 May 2000

http://www.roaringpenguin.com
dfs@roaringpenguin.com

TCP/IP and Linux: Overview

- TCP/IP Basics:
  - Networks and the Internet
  - IP Addresses
  - Protocols: IP, UDP, TCP, ICMP
  - Routing
  - DNS
  - Encapsulation: Ethernet, PPP
  - Application Protocols: FTP, Telnet, SMTP, HTTP

TCP/IP and Linux: Overview (2)

- TCP/IP and Linux:
  - Configuring Ethernet Interfaces
  - Configuring PPP Interfaces
  - Routing
  - DHCP
  - DNS
  - Network Services
  - Diagnostic tools: ping, nslookup, traceroute

TCP/IP and Linux: Overview (3)

- Security
  - Firewalls
  - Proxies
  - Network−based attacks
  - SSH
- Questions and answers

Networks and the Internet

- A *network* is a collection of computers which can communicate directly with one another.
- A *protocol* is an agreed−upon method two computers use to communicate.
- An *internet* is a set of networks connected to one another via *routers*.
- The *Internet* is the world−wide internet of systems which use the TCP/IP protocols

Layers

- Protocols are usually arranged in *layers* with the resulting suite called a *protocol stack*.
- *Physical Layer:* Electronics and wire.
- *Datalink Layer:* Software to get data directly to another computer.
- *Network Layer:* Software to route data possibly through multiple computers.
- *Transport Layer:* Software which provides end−to−end communication services.

IP Addresses

- Every computer (actually, interface) on the Internet is assigned an *address*.
- An *IP Address* is a 32−bit binary number, usually written as four dot−separated decimal numbers ranging from 0 to 255.
- Examples of IP addresses: 192.168.5.3, 134.117.9.94, 127.0.0.1

Private IP Addresses

- Some IP addresses are *reserved* for private use. You should never see these addresses on the real Internet.
- Reserved addresses are:
  - 10.0.0.0 through 10.255.255.255
  - 172.16.0.0 through 172.31.255.255
  - 192.168.0.0 through 192.168.255.255

Network Addresses

- Hosts on a single network are assigned IP addresses within a contiguous range.
- For example, the network of addresses beginning with 192.168.1 encompasses 256 IP addresses. It is often written as 192.168.1.0/24.
- The ***/24*** means that the 24 most−significant bits define the *network address*. The remaining bits are the *host address*.

Network Addresses (2)

- Networks can be split at any position. For example, 10.2.3.128/28 consists of the sixteen addresses 10.2.3.128 through 10.2.3.143.
- Rather than a bit *count*, network addresses can be written as an address and a mask consisting of *count* 1−bits:
  - 192.168.1.0/24 = 192.168.1.0/255.255.255.0
  - 10.2.3.128/28 = 10.2.3.128/255.255.255.240

Protocols: IP

- The *TCP/IP Protocol Suite* consists of a number of layered protocols.
- The *Internet Protocol* (IP) is a best−effort network protocol. It attempts to deliver *packets* of information from a source computer to a destination computer, but makes no guarantee that packets will arrive in order, unduplicated or at all.
- IP is a bit like the postal system

Protocols: UDP

- The *User Datagram Protocol* (UDP) is a simple transport protocol built on IP. Like IP, it is best−effort and unreliable.
- UDP allows a *process* on one computer to send packets to a *process* on another. It adds *port numbers* to the IP address to distinguish processes.
- UDP is used where simplicity is essential and for broadcasting/multicasting.

Protocols: TCP

- The *Transmission Control Protocol* (TCP) is a complex, reliable, stream−oriented transport protocol built on IP.
- A TCP connection lets one process send a stream of data and ensures that the other end receives the exact same stream.
- TCP is used for many applications such as e−mail transport, Web browsing and FTP.

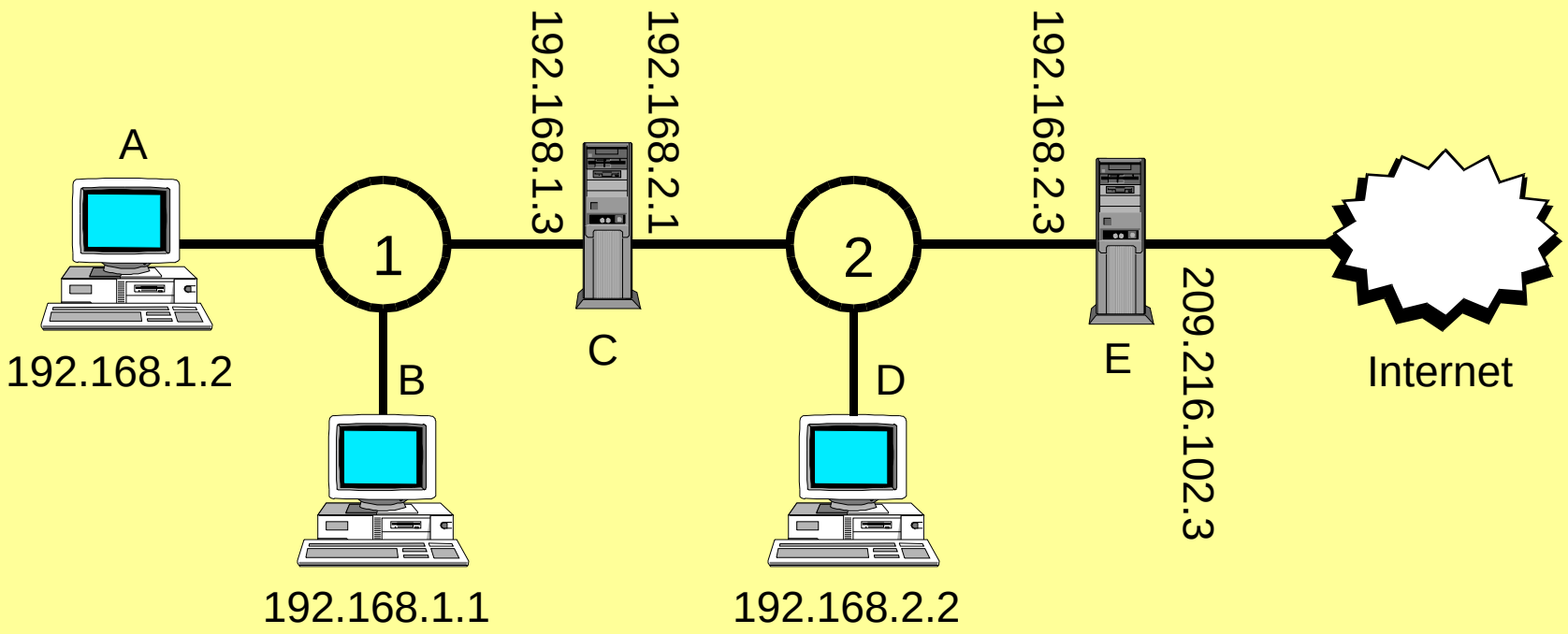
Protocols: ICMP

- The *Internet Control Message Protocol* (ICMP) controls the operation of the Internet itself.
- It reports on congestion, unreachable hosts, changed routes, and so on.
- ICMP is used by some programs such as *ping* and *traceroute*. Usually, however, the TCP/IP code in the kernel deals with ICMP messages.

Routing

- TCP/IP allows packets to be routed across many machines.
- A machine with more than one interface can be configured as a *router*. It passes packets between two (or more) different networks.
- A router has a *routing table* which determines how packets are routed. Routing decisions are usually based on destination IP address, but can take into account other parameters.

Routing Example

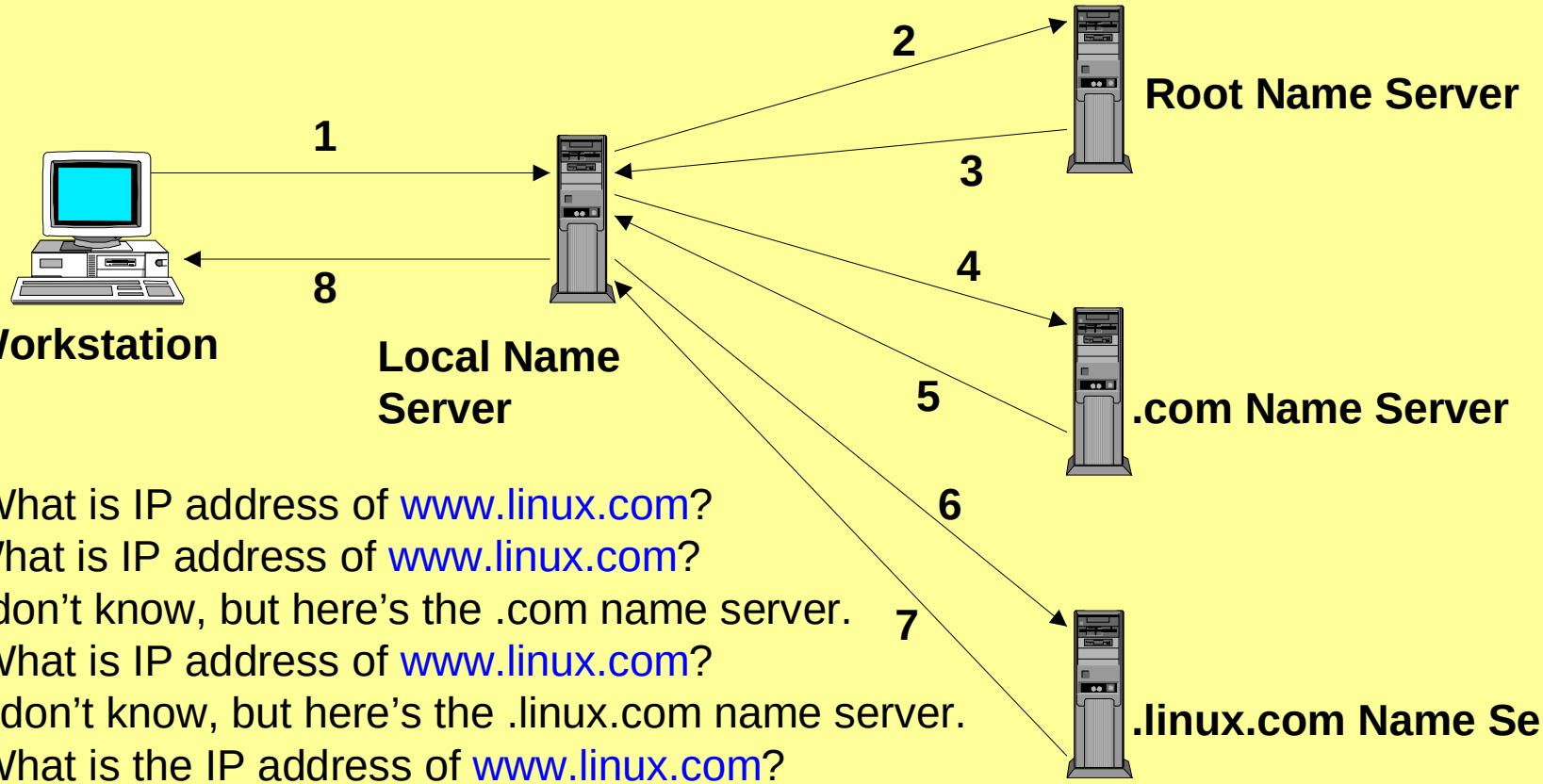


Packets from A to B are sent directly.
Packets from A to D are sent through C.
Packets from A to an Internet host are sent through C and then E.

DNS (Domain Name Service)

- Remembering numerical IP addresses is hard. Humans prefer to name machines.
- The DNS is a *distributed, hierarchical database* which maps machine names (e.g. **www.roaringpenguin.com** to IP addresses (e.g. **209.87.224.131**)
- There are several *root DNS servers* which refer queries to DNS servers for subdomains.

Example DNS query



1) What is IP address of www.linux.com?
2)What is IP address of www.linux.com?
3)I don't know, but here's the .com name server.
4) What is IP address of www.linux.com?
5) I don't know, but here's the .linux.com name server.
6) What is the IP address of www.linux.com?
7) It is 198.168.203.55
8) It is 198.168.203.55

Local name server now *caches* this information. Subsequent lookups do not invoke queries 2 through 7 until cached data expires.

Encapsulation

- Computer sends a chunk of data:

- Transport layer adds transport header:

- IP layer adds IP header:

- Physical layer adds header/trailer:

- Ethernet adds source and destination Ethernet addresses, protocol field and checksum.
- PPP adds framing bytes, protocol field and checksum.

Encapsulation (2)

- Most common datalink layers are Ethernet and PPP. Less common are token ring and PPPoE (PPP over Ethernet).
- Ethernet is a broadcast medium typically used on LANs.
- PPP (Point−to−Point Protocol) allows IP (and other protocols) to be used over a serial link, typically used to connect a LAN to an ISP.

Applications: FTP

- FTP (File Transfer Protocol) is used to transfer files across the Internet. FTP uses two TCP connections: A control connection and a data connection.
- FTP operates in two modes: *active*, in which the server initiates the data connection, and *passive*, in which the client does.
- Active−mode FTP has implications for firewalls (more later).

Applications: Telnet

- Telnet is used for remote interactive logins.
- The Telnet client can be used to debug other protocols (you can run an "interactive" HTTP session, for example.)
- Telnet is old and *insecure*: Login names and passwords are transmitted across the Internet in cleartext.
- Do not use Telnet for remote access; use the Secure Shell instead (more later.)

Applications: SMTP

- SMTP (Simple Mail Transfer Protocol) is used by mail transfer agents to transmit e−mail across the Internet.
- SMTP is insecure and can easily be spoofed, although extensions for authentication exist and are being implemented.
- E−mail clients typically use SMTP to *send* mail, but another protocol (like POP or IMAP) to *receive* mail.

Applications: HTTP

- Hyper−Text Transfer Protocol (HTTP) is used by Web clients and servers.
- HTTP is a simple TCP−based protocol for retrieving Web documents.
- HTTP is insecure; a separate protocol (HTTPS) provides security (encryption).

TCP/IP and Linux

We switch focus from TCP/IP fundamentals to the Linux implementation.

Configuring Ethernet Interfaces

- Every interface on a Linux system has a name. Ethernet interfaces are called **eth*n***. For example, the first Ethernet interface is called **eth0**, the second is **eth1**, and so on.
- The **ifconfig** command configures an Ethernet interface. To see the settings of eth0, for example, type:

```
ifconfig eth0
```

```
ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:40:05:6E:72:C0
          inet addr:192.168.2.1  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:321687 errors:0 dropped:0 overruns:0 frame:0
          TX packets:549007 errors:0 dropped:0 overruns:0 carrier:0
          collisions:32 txqueuelen:100
          Interrupt:5 Base address:0x300
```

- *Link encap: Ethernet* denotes Ethernet interface
- *inet addr:192.168.2.1* is IP address of Ethernet interface.
- *Bcast:192.168.2.255* is the *broadcast* address of the LAN.
- *Mask:255.255.255.0* is the *network mask*.

IP Address

- Each interface is associated with one IP address.
- This IP address is used as the source address for data originating on the host which flows through the interface.
- You should use private IP addresses for internal LANs. External connections will be assigned IP addresses by your ISP.

Broadcast Address

- Ethernet interfaces have a *broadcast address*. Packets sent to this address are received by *every* host on the local network. The broadcast address is the network address with a host address of all 1−bits.
- Historically, some systems used a host address of all 0−bits for the broadcast address. For this reason, you should not assign a host address of all−1's or all−0's to a real host.

Flags

- UP RUNNING BROADCAST MULTICAST are flags. They specify that the interface is active (UP RUNNING), that it is on a broadcast medium, and that it supports *multicasting*.
- Multicasting is a mechanism for sending data to some hosts (unlike broadcasting, which sends to all hosts on a LAN.) I will not discuss multicasting in this presentation.

MTU

- MTU stands for Maximum Transmission Unit. It specifies the largest IP packet which can be transmitted through the interface.
- Ethernet interfaces almost always have an MTU of 1500 due to hardware limits on the size of Ethernet frames.
- Packets larger than an interface MTU must be *fragmented* and reassembled by the receiver. This adds overhead; fragmentation is undesirable.

Configuring the Interface

- To configure the interface, use:

  `ifconfig eth`*`n ip_addr`* `netmask` *`mask`*

- ifconfig can often deduce the correct netmask from the IP address, and can usually deduce the correct broadcast address.
- ifconfig has *many* other options; see the man page. For previous example:

```
ifconfig eth0 192.168.2.1 netmask 255.255.255.0
```

Configuring PPP Interfaces

- PPP interfaces are more complicated to configure than Ethernet interfaces.
- PPP interfaces rely on the *PPP Daemon* (pppd) to set up and configure the interface. Actual data transfer is done by the kernel.
- *pppd* has many options; read the man page.
- PPP is a complex protocol which allows for negotiation of IP addresses, compression techniques, and so on.

Configuring PPP Interfaces (2)

- Once a PPP interface is up, you can look at it with **ifconfig**:

```
ppp0  Link encap:Point-to-Point Protocol
      inet addr:216.209.152.27  P-t-P:216.209.152.1 Mask:255.255.255.255
      UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1492  Metric:1
      RX packets:14 errors:0 dropped:0 overruns:0 frame:0
      TX packets:13 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:10
```

- Note the flag POINTOPOINT and the P–t–P IP address. This is the IP address of the PPP peer.
- PPP interfaces are named **ppp*n***.

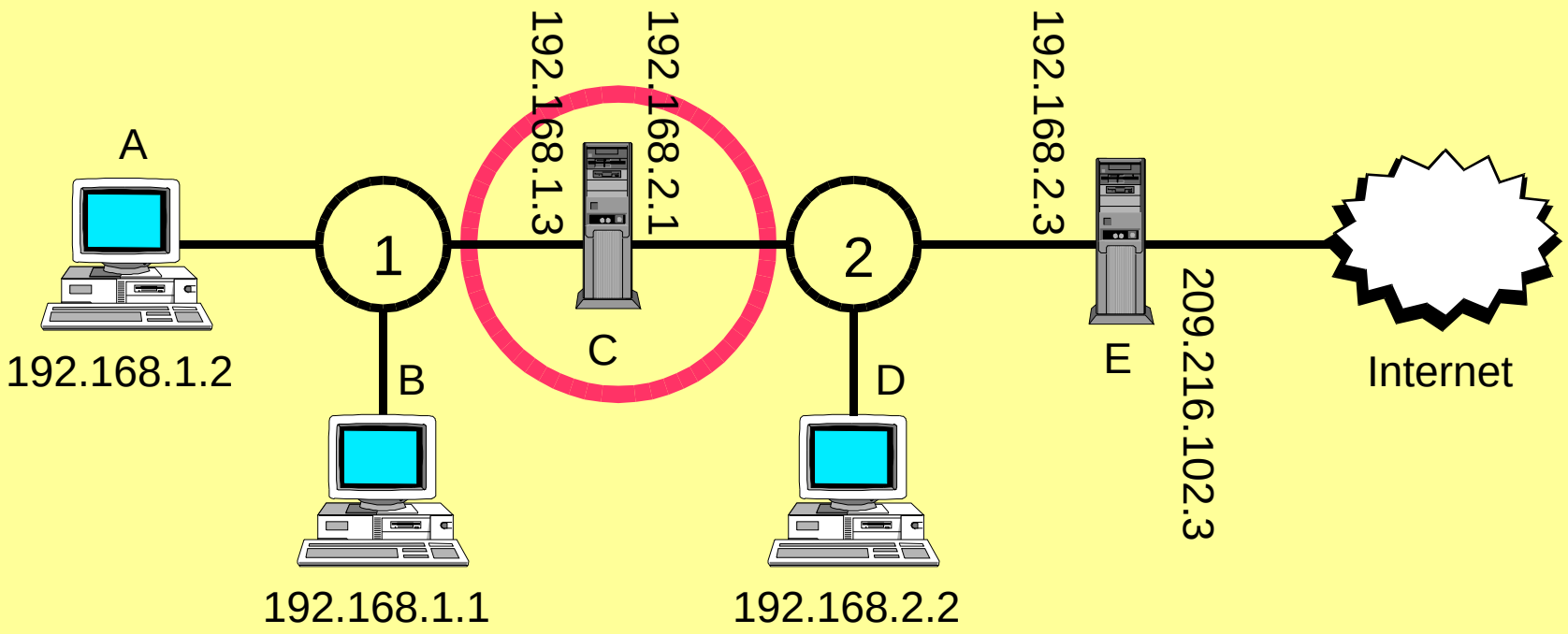
Routing

- Routing consists of deciding which *interface* to send a packet out of, and which *gateway* (if any) to send it to.
- A packet destined for a host on the local network is sent out of the local interface directly to the destination.
- A packet destined for a host not on the local network is sent to a router. If no router exists on the local network, the packet cannot be routed and is dropped.

Routes

- A *host route* specifies which interface to use for a specific host.
- A *network route* specifies which interface (and possibly gateway) to use for all hosts on a given network.
- A *default route* specifies which interface and gateway to use if no other route exists.
- The collection of routes is called the *routing table*.

Routing Table Example



Consider C's routing table:

- Packets for network 1 go out of the left interface.
- Packets for network 2 go out of the right interface.
- All other packets go out of the right interface through the gateway E.

C requires two network routes and a default route.

Adding and Deleting Routes

- The **route** command adds and deletes routes. Also, the **ifconfig** command automatically adds appropriate network routes when you configure an Ethernet device.
- The **pppd** daemon can set up default routes (the most common case.)
- Example: **`route add default gw 216.209.1.32`**
- See **route** man page for details.

Checking the Routing Table

- Use **route −n** to print the routing table:
- Example:

```
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
216.209.153.1   0.0.0.0         255.255.255.255 UH    0      0        0 ppp0
192.168.2.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
127.0.0.0       0.0.0.0         255.0.0.0       U     0      0        0 lo
0.0.0.0         216.209.153.1   0.0.0.0         UG    0      0        0 ppp0
```

- First line is a host route to 216.209.153.1.
- Second line is a network route to 192.168.2.0/24.
- Third line is local loopback route.
- Fourth line is default route.

DHCP

- In a large LAN, it is annoying to have to assign IP addresses to each workstation by hand.
- The Dynamic Host Configuration Protocol (DHCP) allows you to centralize the administration of IP addresses.
- A central DHCP server is responsible for assigning IP addresses to hosts.

How DHCP Works

- When a host boots, it *broadcasts* a DHCP discovery packet. This is a UDP packet.
- One or more DHCP servers reply with *offers*.
- The client selects a DHCP server and sends it a DHCP request.
- The server replies and informs the client of its IP address, network mask, default gateway, and so on.

DHCP Under Linux

- Most Linux distributions have simple "point−and−click" GUIs for configuring an interface as a DHCP client. Internally, they use a program called **pump** or **dhcpcd** to control the DHCP requests.
- A program called **dhcpd** lets Linux act as a DHCP server. It uses a plain−text configuration file which lets you specify the available pool of IP addresses, network mask, and so on.

Sample dhcpd Configuration File

```
# Configuration file for DHCP server.
# Please see /usr/doc/dhcpd-1.0pl2 for examples,
#  and view the manpage dhcpd.conf(5).

subnet 192.168.1.0 netmask 255.255.255.0 {
    option domain-name          "my-lan.com";
    option ip-forwarding        0;
    option subnet-mask          255.255.255.0;
    option domain-name-servers 192.168.1.254;
    option routers              192.168.1.254;
    option broadcast-address    192.168.1.255;
    range                       192.168.1.100 192.168.1.200;
    default-lease-time          86400;
    max-lease-time              86400;
}
```

DNS

- Linux is automatically set up to be a DNS client. Simply add the addresses of the name servers in the file /etc/resolv.conf. For example:

```
domain roaringpenguin.com
nameserver 192.168.3.2
nameserver 192.168.3.1
```

DNS Server

- The **named** program (part of BIND, the Berkeley Internet Name Daemon) is the standard Linux DNS server.
- Configuring **named** can be rather tricky; consult the online documentation.
- Each domain which **named** knows about is called a *zone*.
- The master file /etc/named.conf lists all the zones as well as global **named** options.

Zones

- A *master zone* is one for which this server is authoritative. All of the DNS data (*zone records*) are held in a file on the server.
- A *slave zone* is one for which this server is a backup server. The master data is periodically transferred from the primary server.
- A *hint zone* is one about which this server knows nothing, but for which it caches DNS answers.

DNS Records

- The DNS system uses several *record types*:
  - **A** records map host names to IP addresses.
  - **PTR** records map IP addresses to host names.
  - **MX** records specify which machines accept e−mail for the specified domain.
  - **NS** records list the name servers for a specified domain.
  - **CNAME** records list "canonical" names for nicknames.

DNS Zone Files

- DNS zone data are stored in special text files. A zone file simply consists of a list of DNS records.
- The format of the zone files is described in BIND’s online documentation.
- For more details, read "DNS and BIND, 3rd Edition" by Paul Albitz and Cricket Liu, O’Reilly and Associates.

Network Services

- Linux comes with many network services, including:
  - TELNET (the **telnetd** program) for remote login.
  - Apache for HTTP (Web) service
  - The **finger** service for finding people.
  - The FTP service.
  - The **talk** service for real−time chat.
  - The **daytime** service for getting the time−of−day.
  - The POP−3, IMAP and SMTP mail services.
  - The NFS service for UNIX file sharing.
  - The SMB service for Windows file and printer sharing.

Controlling Network Services

- Some network services (HTTP, SMTP) are typically started at boot time and continue to run until the computer is shut down.
- Many other services are controlled by the **inetd** program. This program "listens" on many Internet ports. When an incoming connection arrives, it starts the correct program for the specified service. This eliminates the need to have many running servers, especially for seldom−used services.

Configuring inetd

- The text file /etc/inetd.conf lists the services controlled by **inetd**. Read the manual page for details.
- You should *disable* all services except those you really need. Any running network service is a potential security risk; minimize the danger by turning off unnecessary services.

Diagnostic Tools: ping

- The **ping** program sends an ICMP "Echo−Request" message.
- Most hosts respond with an ICMP "Echo−Reply" message.
- **ping** prints the replies as they come in, as well as the round−trip time.
- **ping** is useful to verify basic network connectivity on a LAN.

Diagnostic Tools: traceroute

- The **traceroute** program attempts to find the route packets take to a particular destination. It works by playing tricks with UDP packets and may not be absolutely reliable.
- Sample **traceroute** output:

```
traceroute to 209.217.112.242
 1  shevy.roaringpenguin.com  46.326 ms  62.678 ms  64.553 ms
 2  206.108.100.5  17.654 ms  18.212 ms  14.626 ms
 3  206.108.100.129  18.195 ms  18.270 ms  19.228 ms
 4  206.108.100.140  346.651 ms  200.824 ms  17.476 ms
 5  206.47.214.202  57.968 ms  41.694 ms  36.103 ms
 6  core2-vlan3.magma.ca  36.858 ms  26.085 ms  18.501 ms
 7  core1-vlan25.magma.ca  21.178 ms  18.803 ms  25.615 ms
 8  border6-faste0-0.magma.ca  26.172 ms  26.218 ms  30.726 ms
 9  209.217.112.242  31.639 ms  25.466 ms  32.078 ms
```

Diagnostic Tools: nslookup and dig

- The **nslookup** program lets you make DNS queries interactively.
- You can use it to verify the correctness of your DNS setup, and to look for various types of DNS records.
- The **dig** program performs similar queries, but is command−line driven (not interactive) and easier to use in scripts.

Network Security

- Connecting a LAN to the Internet poses ***serious*** security risks.
- Linux and UNIX were designed to be as easily used remotely as locally. While this is very powerful and convenient, it can also be devastating if a computer is compromised.
- Compromised Linux machines can easily be used to launch attacks on other computers, perhaps even making you liable for damages.

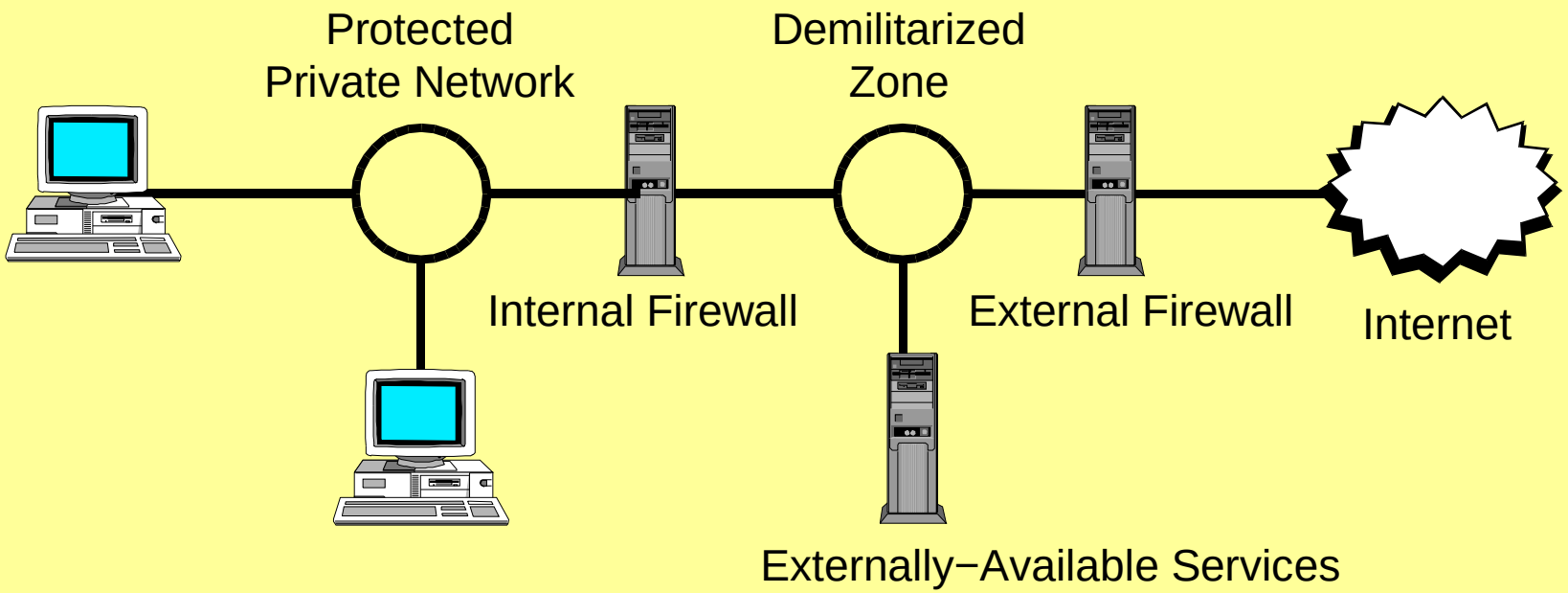
Network Security (2)

- With the rise of DSL and cable "always−on" high−speed connections, the number of machines available for compromise is skyrocketing.
- Network security must be taken very seriously. Even if you don’t have anything important on your computer, you owe it to others to secure it so it cannot be used to launch attacks.

Firewalls

- A firewall is a router which connects a protected LAN to the outside world.
- The firewall inspects all packets passing through it and rejects those which are considered "unacceptable."
- Firewalls can only make decisions based on source and destination IP addresses and ports and some other information in the IP packets. Some firewalls maintain state to track TCP connections.

Firewall Example



Firewalls (2)

- Firewalls often provide Network Address Translation (NAT) which lets you hide a LAN of private IP addresses behind a single routable IP address.
- Firewalls offer protection against unauthorized access to internal services.
- Firewalls **do not** protect against bugs in those internal services which are exposed to the Internet.

Proxies

- Whereas firewalls operate at the network level (with some transport−level facilities), proxies operate at the application level.
- For each type of application (HTTP, FTP, etc.) a separate proxy program is needed.
- Proxies allow much finer−grained decision−making. They can allow or block connections based on user ID's, time−of−day, network traffic, etc.
- Proxies do not allow *any* external packets directly into the internal network.

Proxies (2)

- The **squid** caching proxy is a popular Linux proxy.
- It not only controls network access, but also caches Web and FTP pages for improved bandwidth utilization.
- **squid** is highly−configurable and features flexible rules to allow or deny access.

Network−Based Attacks

- Some network−based attacks exploit bugs in TCP/IP implementations. Examples: The "ping of death" whereby a malformed Echo−Request packet crashed a host.
- To reduce the possibility of these attacks, firewall packets aggressively. Do not allow ICMP Echo−Request packets in; perform defragmentation on the firewall.

Network−Based Attacks (2)

- Most network−based attacks exploit bugs in service implementations. Examples: The **wu−ftpd** FTP program contained numerous bugs which allowed attackers to get a root shell on the victim.
- To reduce the chances of these kinds of attacks, do not run services unless they're really necessary. Upgrade buggy services promptly. Consider tools like StackGuard to make bug exploits more difficult.

Network−Based Attacks (3)

- Some network−based attacks rely on "password−sniffing" or eavesdropping.
- To reduce the chances of these attacks, do not use protocols which transmit clear−text passwords (TELNET, POP−3). Use more secure replacements.
- There are some tools which attempt to find eavesdropping hosts on the LAN. Run them periodically if you don’t trust all your LAN hosts.

Network−Based Attacks (4)

- There are other network−based attacks such as "man−in−the−middle" attacks which require more sophistication than the other attacks and are quite uncommon.
- If you want to have secure communication between two networks over the Internet, consider an IPSec implementation for Linux like FreeS/Wan.

SSH

- The Secure Shell (SSH) provides secure remote access for Linux machines. Use it instead of Telnet, rsh or rlogin.
- The OpenSSH package (www.openssh.com) is a free implementation of SSH for UNIX and Linux.
- Patent issues may prevent its use in the United States until RSA's public−key encryption patent expires on 20 September, 2000.

SSH (2)

- SSH uses public−key cryptography for authentication and symmetric encryption for privacy.
- This means that an eavesdropper cannot sniff your password, and cannot determine the contents of your session, even if he captures every packet.
- SSH is easy to use and install. It should be required on every Linux system, and Telnet should be banished!

Questions and Answers