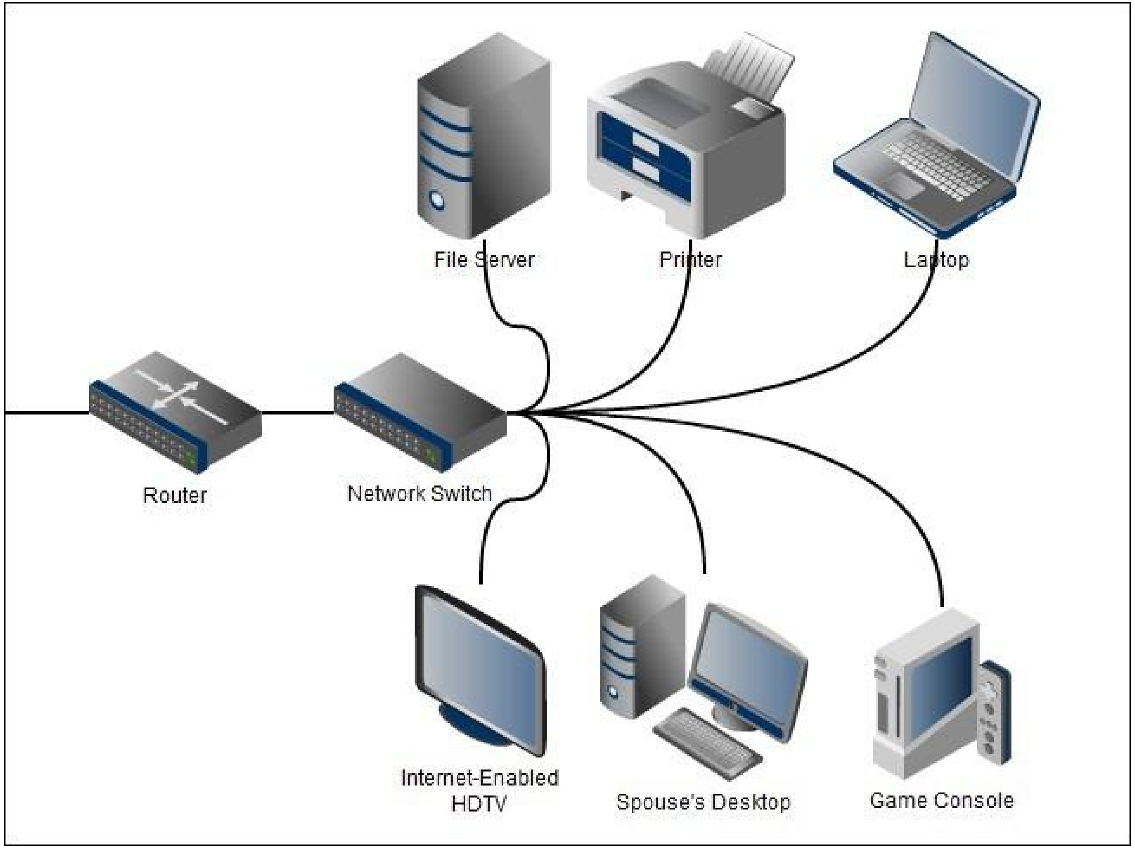
Linux Networking

What is a network?

- A collection of devices connected together
- Can use IPv4, IPv6, other schemes
- Different devices on a network can talk to each other
- May be walls to separate different networks



Terminology Overview

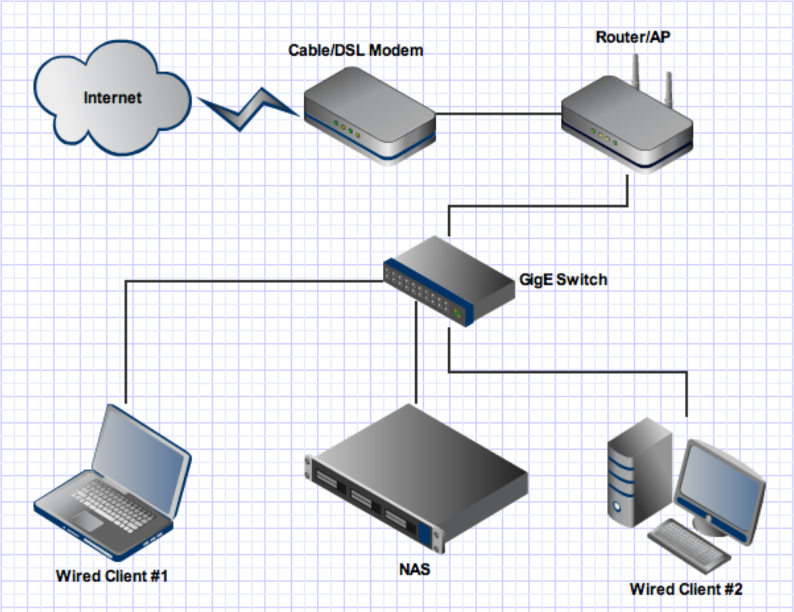
- IP Address: the per-network unique identifier used to find you on a network
    - IPv4: 4 bytes, typically shown as decimal numbers, e.g. 123.234.321.003
        - Groups of IPv4 addresses are shown using /X notation, where X is fixed bits
        - 192.168.0.0/16 means 192.168 is fixed, but the 0’s can be any number 0-255
    - IPv6: 16 bytes, shown as hex numbers, e.g. 3614:DEAD:BEEF:4A7C:3614:::4A7C
- MAC Address: a globally unique identifier given to every network interface
    - 12 bytes, shown as hex numbers, e.g. dc:0e:a1:f3:fb:99
    - Not supposed to be changeable (but can be changed)
- Subnet: an isolated network separated from the global internet
- MTU: the maximum packet size sent from your interface
- Gateway: a device that routes traffic between two or more networks
- NAT: network address translation, used to allow devices to share IPs
- DHCP: a protocol for requesting ip addresses from a gateway
- Subnet Mask: a bitmask for finding the part of an ip is in subnet
    - e.g. 255.255.255.0: IP 192.168.1.123 masked by 255.255.255.0 equals 0.0.0.123

The Internet and Networks

- Internet was originally intended to have all devices publicly addressable via IPv4, without need to wall off certain sections
- Privacy, security, and IPv4 exhaustion concerns led to the creation of isolated networks
- Individuals on isolated networks needed methods to contact the outside world via a single public IP address
- Private networks can use 192.168.0.0/16, 10.0.0.0/8, and 172.16.0.0/12
  - These ip addresses can NEVER be used for a public facing IP
  - All other ip addresses can be actual websites/destinations, so they CANNOT be used for personal networks

NAT and Subnetting

- Your home/work network will use a gateway to contact the outside world
- Your private network will have its own subnet, typically 192.168.1.0/24
  - This means that your computer can directly contact anything in that subnet, but must go through the gateway to contact anything else
- The router performs Network Address Translation (NAT) to keep track of what packets go to which computer on the subnet



Networking and Linux

- Most non-server distros include a connection manager, such as
  - Network-manager (gui, Most common)
  - Netctl (cli, Arch Linux)
  - wicd (cli)
- Manual configuration is always available, typically using
  - ifconfig (most common, legacy)
  - ip (newer)
- DHCP can be performed using:
  - dhclient (most common)
  - dhcpcd
- Make sure to stop/disable the connection manager when trying to do manual configuration (service network-manager stop or similar)

DHCP Network Connection

- The easiest way to connect to a network is with DHCP
- Most computers can be set up using:
  - dhclient <device>
  - dhcpcd <device>
- This will get the machine and ip, setup the subnet mask, and find DNS servers

IFCONFIG/IP Common Commands

- Show connection status: *ifconfig <device>* or *ip addr show <device>*
    - Device is the name of the interface (eth0, eth1, wlan0, etc)
    - Can be left blank to show all
- Turn on/off *ifconfig <device> <up/down>* or *ip link set <device> <up/down>*
- Set IP Address:
    - *ifconfig <device> <ip> netmask <netmask>*
    - *ip addr add <ip>/<mask> dev <device>*
    - A netmask of 255.255.255.0 is the same as /24, 255.0.0.0 is /8
- Set Gateway:
    - route add default gw <gateway ip>
    - ip route add default via <gateway ip> dev <device>

DNS Setup

- DNS is used to find the ip address of a web service based on its text name
  - e.g. purdue.edu -> 128.210.7.200
- DNS Servers must be added in order to be able to use text website names instead of IPs
- Linux looks for DNS servers in /etc/resolv.conf
- For simple setup, simply write, one per line: *nameserver <ip>*
- To add a search path, write: *search <domain>*
  - A search path of *purdue.edu* would make typing *mypurdue* into your browser automatically check mypurdue.purdue.edu
- resolv.conf is automatically rewritten anytime you run a DHCP client or when your connection manager runs a DHCP client.
- Purdue’s DNS servers are: 128.210.11.5, 128.210.11.57
- Engineering Computer Network’s DNS is: 128.46.154.76

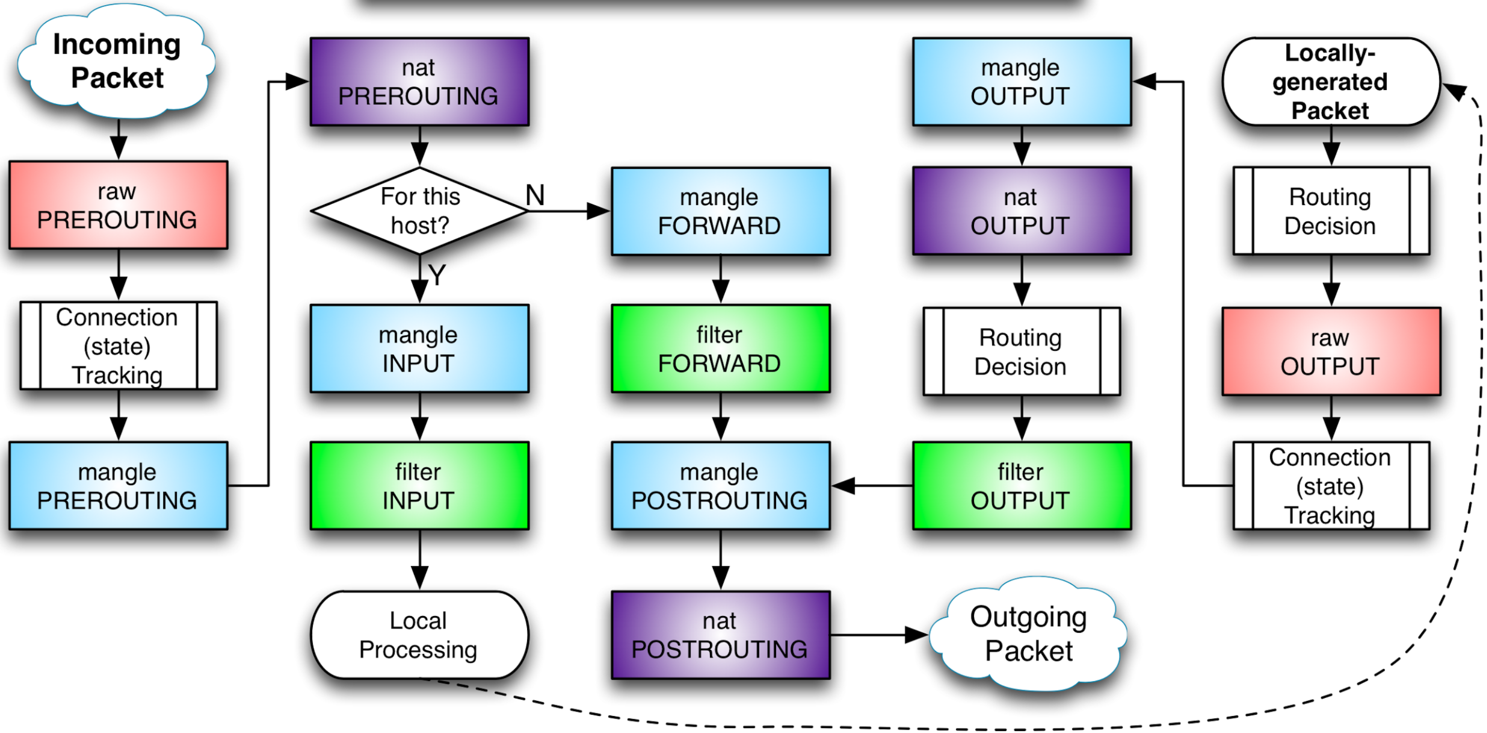
Static IP Setup

- If you need a simple static ip setup, files can be edited to save the setting between reboots
- Debian Based: /etc/network/interfaces
- Fedora Based: /etc/sysconfig/network **and** /etc/sysconfig/network-scripts/ifcfg-<device>
- Archlinux: use netctl or write a startup script that calls ip
- Quick setup details available at http://goo.gl/8jEKSe

Firewalls

- Firewalls allow certain computers to access network services on your device while prohibiting others
- The linux kernel contains iptables, which is used for all firewalls
- Several software packages exist to generate and manage iptables
- Simple setups can be done using scripts that run on startup

iptables Process Flow



Created by Phil Hagen (ver 2014-09-25)
for SANS FOR572: Advanced Network Forensics and Analysis
See http://sans.org/for572 for more information

Derived from : http://www.iptables.info/en/structure-of-iptables.html

iptables simple firewall

- iptables follows the following syntax (many variations exist):
  - iptools -A <chain> -p <protocol> <filters> -j <destination chain>
  - Default chains are INPUT, OUTPUT, FORWARD, and ACCEPT. Custom chains can be made.
  - protocols can be tcp, udp, or icmp
  - filters can be things like --dport 22 (destination port 22)
- A Simple firewall that allows only SSH could be:
  - iptables -P FORWARD DROP; iptables -P OUTPUT ACCEPT; iptables -P INPUT DROP
  - iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
  - iptables -A INPUT -i lo -j ACCEPT
  - iptables -A INPUT -p tcp --dport 22 -j ACCEPT
- A more robust simple stateful firewall example is at https://wiki.archlinux.org/index.php/Simple_stateful_firewall