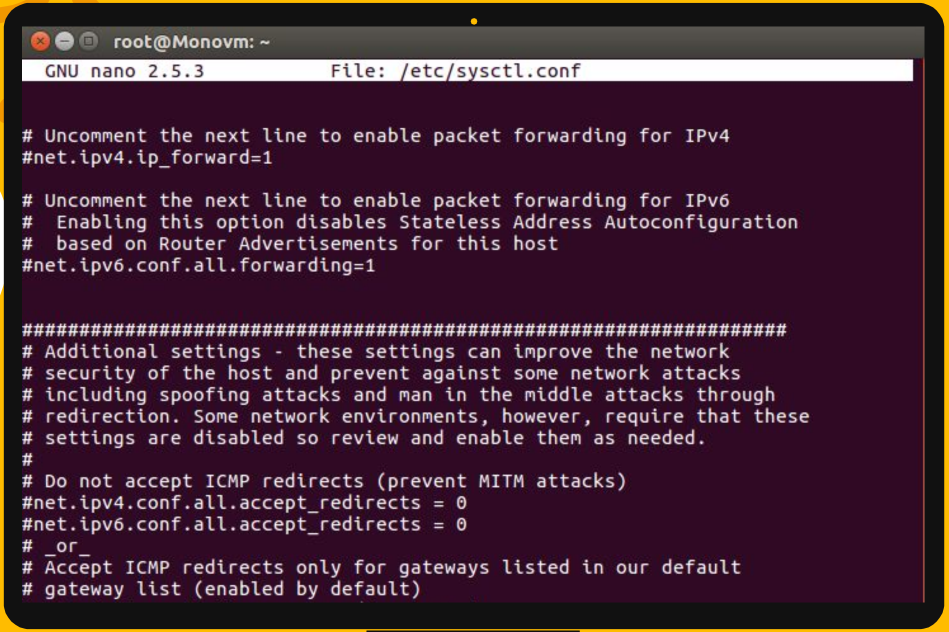
Networking Configurations



```
  GNU nano 2.5.3           File: /etc/sysctl.conf


# Uncomment the next line to enable packet forwarding for IPv4
#net.ipv4.ip_forward=1

# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1


###################################################################
# Additional settings - these settings can improve the network
# security of the host and prevent against some network attacks
# including spoofing attacks and man in the middle attacks through
# redirection. Some network environments, however, require that these
# settings are disabled so review and enable them as needed.
#
# Do not accept ICMP redirects (prevent MITM attacks)
#net.ipv4.conf.all.accept_redirects = 0
#net.ipv6.conf.all.accept_redirects = 0
# _or_
# Accept ICMP redirects only for gateways listed in our default
# gateway list (enabled by default)
```

/etc/sysctl.conf

General networking security settings for the kernel in here

- ✘ General syntax is [option] = 0 or 1
- ✘ klaver.it is the move

- Many options
  - IPv4 TCP SYN cookies (DoS attaccs)
    - net.ipv4.tcp_syncookies
  - Preventing IP spoofing attaccs
    - net.ipv4.conf.all.rp_filter
  - IPv4 TCP SYN,ACK retries
    - net.ipv4.tcp_synack_retries
  - IPv4 forwarding
    - net.ipv4.ip_forward

- ✗ IPv4 TIME-WAIT assassination protection enabled
  - ■ net.ipv4.tcp_rfc1337
- ✗ IPV4 sending ICMP redirects
  - ■ net.ipv4.conf.all.accept_redirects
  - ■ net.ipv4.conf.default.accept_redirects
  - ■ net.ipv4.conf.all.secure_redirects
  - ■ net.ipv4.conf.default.secure_redirects
  - ■ net.ipv6.conf.all.accept_redirects
  - ■ net.ipv6.conf.default.accept_redirects
- ✗ Most secure ASLR enabled
  - ■ kernel.randomize_va_space
- ✗ Ignore broadcast ICMP echo requests
  - ■ net.ipv4.icmp_echo_ignore_all
- ✗ IPv4 accept source routing
  - ■ net.ipv4.conf.[all/default].accept_source_route
- ✗ IPv6 disabled
  - ■ net.ipv6.conf.[all/default].disable_ipv6

/etc/resolv.conf

- Allows certain DNS servers to connect to host; can be used to fix DNS issues
- Syntax:
  - nameserver [IP address]
- Tip: Can use addresses to check your Internet conxns using ping cmd
- Make sure you’re not connected to malo DNS server
  - Google DNS servers 8.8.8.8, 8.8.4.4,

Ssave ur sysctl.conf

Sysctl -ep

/etc/resolvconf/resolv.conf.d

The folder itself

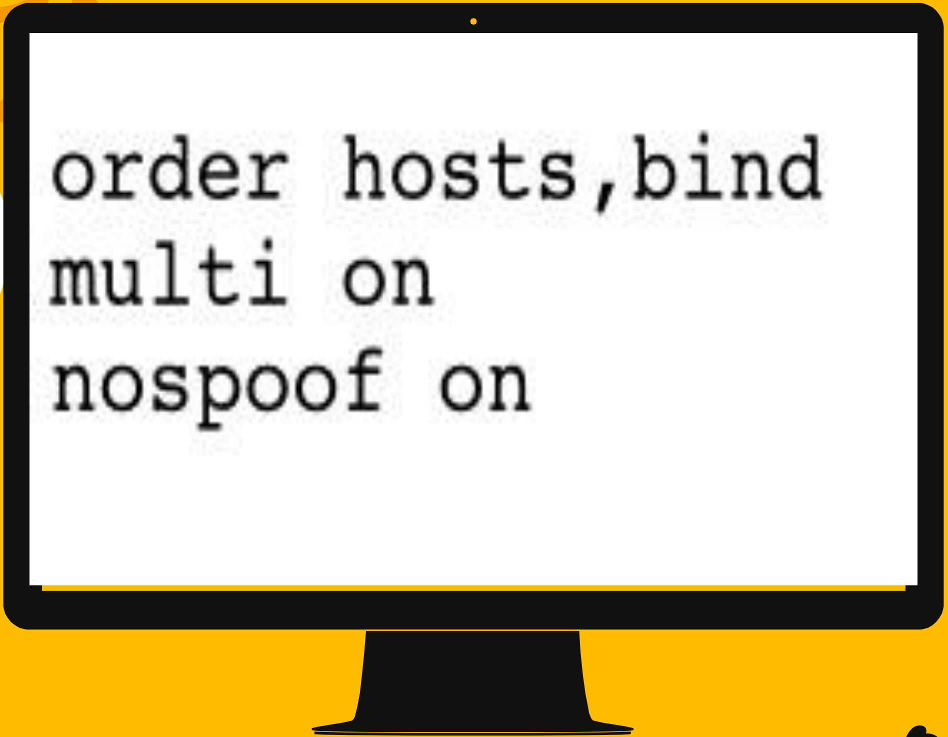
Contains other configuration files for resolving

./base

File containing basic resolver information. The lines in this file are included in the resolver configuration file even when no interfaces are configured.

/etc/hosts

- ✘ Contains hosts that can be contacted without a name service (ex. DNS); creates aliases for IP addresses
- ✘ Always has the loopback address 127.0.0.1 (aka localhost)
  - ✗ Needed to test server functionality
  - ✗ Ex: using loopback to test if Apache Guacamole is connecting to remote client



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

/etc/host.conf

Determines how hostnames are resolved

1. Order in which IP addresses are looked up
2. Multiple addresses can be read in hosts file
3. IP Spoofing

/etc/hosts.deny

Blacklisting IP addresses that are not allowed to connect to the host

Firewall

- ✗ Firewall protection has been enabled
  - ■ The configs for these are in personalized file (/etc/ufw/sysctl.conf)

Let's review some concepts

sysctl.conf

General kernel + networking settings

resolv.conf

Determines what DNS servers can connect to host. Other files are in /etc/resolvconf/resolv.conf.d

hosts

Which hosts do not need to be searched for by DNS

host.conf

Configuration for resolving hostnames

hosts.deny

Blacklist of hosts that computer will not resolve

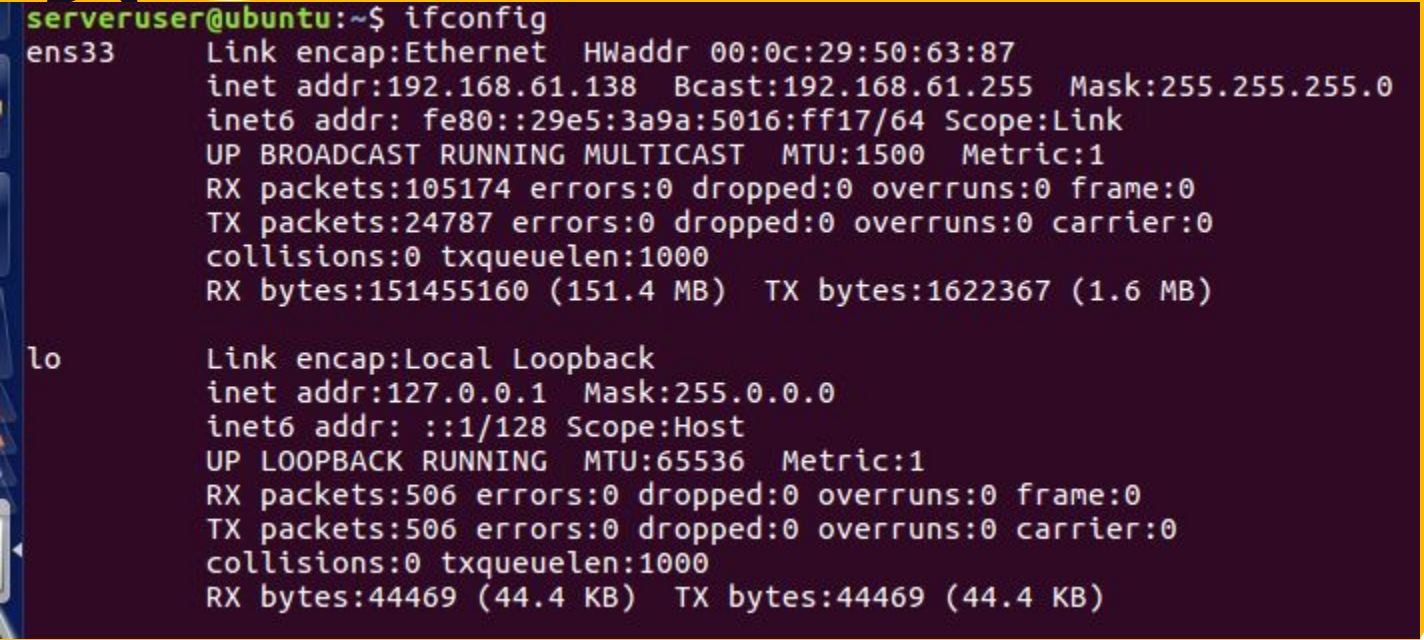
Networking cmds

yeeeee

ifconfig

- interface config
  - interface means your network interface card
- shows information such as your ip addr

ifconfig cont



```
serveruser@ubuntu:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:50:63:87
          inet addr:192.168.61.138  Bcast:192.168.61.255  Mask:255.255.255.0
          inet6 addr: fe80::29e5:3a9a:5016:ff17/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:105174 errors:0 dropped:0 overruns:0 frame:0
          TX packets:24787 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:151455160 (151.4 MB)  TX bytes:1622367 (1.6 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:506 errors:0 dropped:0 overruns:0 frame:0
          TX packets:506 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:44469 (44.4 KB)  TX bytes:44469 (44.4 KB)
```

- ens33 and lo = interfaces
- HWaddr = MAC addr
- inet addr = your dynamic IPv4 addr
    - 127.0.0.1 is reserved for lo
- Bcast = broadcast
- Mask = subnet mask
- inet6 addr = your dynamic IPv6 addr
- Everything else is statistics

Just some reminders

- When doing networking labs, check that you and your partner are on the SAME SUBNET
  - We are all using private IPs behind PAT!!
  - Check inet addr and mask in ifconfig

ping

- Sends echo request to an IP addr
  - if you receive packets back from the IP, then you have network connectivity
- Can check network connectivity during rounds
  - Ex: ping 8.8.8.8 (see if you can reach google)

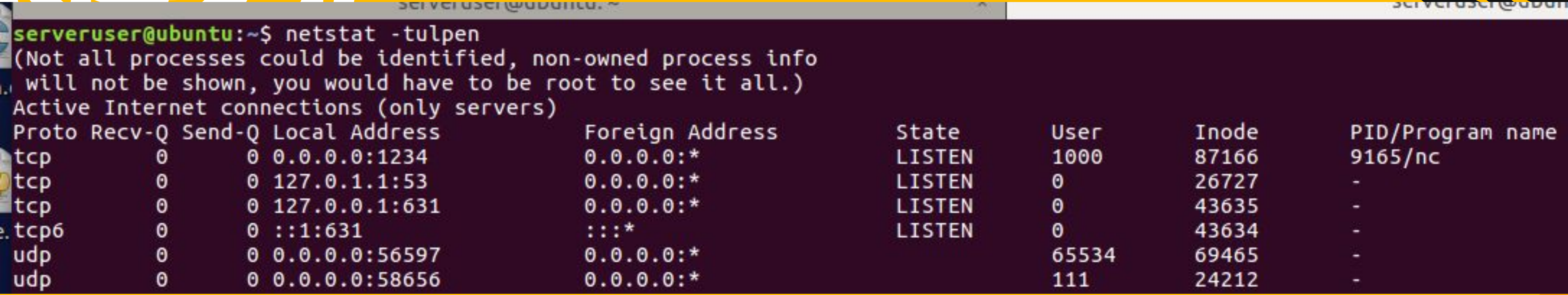
traceroute

- Similar to ping, except it shows each hop
  - Will show whether you have network connectivity + path to get to the IP specified

netstat

- Network statistics
- Tons of options, but netstat -tulpen is a pretty nice set to use
  - -l is for listening

netstat ex:



```
serveruser@ubuntu:~$ netstat -tulpen
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       User       Inode      PID/Program name
tcp        0      0 0.0.0.0:1234            0.0.0.0:*               LISTEN      1000       87166      9165/nc
tcp        0      0 127.0.1.1:53            0.0.0.0:*               LISTEN      0          26727      -
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      0          43635      -
tcp6       0      0 ::1:631                 :::*                    LISTEN      0          43634      -
udp        0      0 0.0.0.0:56597           0.0.0.0:*                           65534      69465      -
udp        0      0 0.0.0.0:58656           0.0.0.0:*                           111        24212      -
```

- This is the result of running netstat -tulpen after setting up nc -l 1234
  - Proto = protocol (tcp or udp)
  - Local address = [IP]:[port number]
  - Foreign address 0.0.0.0 means all IPs
  - PID/program name

nslookup

- Allows you to parse DNS records
  - Ex: nslookup jimmyli.u returns 185.199.111.153

curl/wget

- Retrieve a file from internet
- Ex:
  - curl -O https://wordpress.org/latest.tar.gz
    - -O means redirect stdout to a file
  - wget wordpress.org/latest.tar.gz

nc (Netcat)

- Swiss army knife of networking
  - Simplest connection (can do using 2 terminals on 1 VM)
    - Terminal 1: nc -l 1234
    - Terminal 2: nc localhost 1234
      - Anything entered in on T1 will be outputted to T2, vice versa

nc con't

- Uses:
  - Data transfer
  - Talking to servers
  - Port scanning
- Try using man nc to see some cool nc examples

Pro network troubleshooting tips

- Check connectivity first using ping
  - If ping fails, then you need to take action
- sudo service network-manager restart
- Check your resolv.conf to see that you have a valid DNS
  - 8.8.8.8 always works :D
- Can try renewing IP address using dhclient cmd

Credits

- tdlp.org
- linfo.org
- manpages.ubuntu.com
- Joseph's Divine Wisdom