Step 1: Update System

```
kirbical@kyraserver:~$ sudo apt update
[sudo] password for kirbical:
Hit:1 http://ports.ubuntu.com/ubuntu-ports jammy InRelease
Get:2 http://ports.ubuntu.com/ubuntu-ports jammy-updates InRelease [128 kB]
Get:3 http://ports.ubuntu.com/ubuntu-ports jammy-backports InRelease [127 kB]
Get:4 http://ports.ubuntu.com/ubuntu-ports jammy-security InRelease [129 kB]
Get:5 http://ports.ubuntu.com/ubuntu-ports jammy-updates/main arm64 Packages [2,
831 kB]
Get:6 http://ports.ubuntu.com/ubuntu-ports jammy-updates/main Translation-en [46
9 kB]
Get:7 http://ports.ubuntu.com/ubuntu-ports jammy-updates/main arm64 DEP-11 Metad
ata [112 kB]
Get:8 http://ports.ubuntu.com/ubuntu-ports jammy-updates/main arm64 c-n-f Metada
ta [18.6 kB]
Get:9 http://ports.ubuntu.com/ubuntu-ports jammy-updates/restricted arm64 Packag
es [4,505 kB]
Get:10 http://ports.ubuntu.com/ubuntu-ports jammy-updates/restricted Translation
-en [882 kB]
Get:11 http://ports.ubuntu.com/ubuntu-ports jammy-updates/restricted arm64 DEP-1
1 Metadata [212 B]
Get:12 http://ports.ubuntu.com/ubuntu-ports jammy-updates/restricted arm64 c-n-f
 Metadata [496 B]
Get:13 http://ports.ubuntu.com/ubuntu-ports jammy-updates/universe arm64 Package
s [1,257 kB]
```

```
kirbical@kyraserver:~$ sudo apt upgrade -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
Get more security updates through Ubuntu Pro with 'esm-apps' enabled:
  libwireshark-data libqt5core5a qt5-gtk-platformtheme libwsutil13
  libqt5network5 libqt5dbus5 libqt5widgets5 libwiretap12 wireshark-qt
  libwireshark15 libqt5gui5 libqt5printsupport5 wireshark-common tshark
  wireshark
Learn more about Ubuntu Pro at https://ubuntu.com/pro
The following NEW packages will be installed:
  linux-headers-5.15.0-160 linux-headers-5.15.0-160-generic
  linux-image-5.15.0-160-generic linux-modules-5.15.0-160-generic
  linux-modules-extra-5.15.0-160-generic
The following packages have been kept back:
  libnss-systemd libpam-systemd libsystemd0 libudev1 systemd systemd-oomd
  systemd-sysv systemd-timesyncd udev
The following packages will be upgraded:
  bind9-dnsutils bind9-host bind9-libs cloud-init curl dconf-cli
  dconf-gsettings-backend dconf-service distro-info-data dpkg ghostscript
  ghostscript-x gir1.2-javascriptcoregtk-4.0 gir1.2-webkit2-4.0
  landscape-common libc-bin libc6 libc6-dbg libcurl3-gnutls libcurl4 libdconf1
```

Step 2: Install Snort

```
kirbical@kyraserver:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 00:0c:29:93:0e:21 brd ff:ff:ff:ff:ff:ff
    altname enp2s0
    inet 172.16.167.129/24 metric 100 brd 172.16.167.255 scope global dynamic en
s160
       valid_lft 1770sec preferred_lft 1770sec
    inet6 fe80::20c:29ff:fe93:e21/64 scope link
       valid_lft forever preferred_lft forever
```

```
kirbical@kyraserver:~$ sudo apt install snort -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
snort is already the newest version (2.9.15.1-6build1).
0 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.
```

Step 3: Configure Snort

```
  GNU nano 6.2                          /etc/snort/snort.conf
#--------------------------------------------------
#   VRT Rule Packages Snort.conf
#
#   For more information visit us at:
#     http://www.snort.org                   Snort Website
#     http://vrt-blog.snort.org/    Sourcefire VRT Blog
#
#     Mailing list Contact:      snort-users@lists.snort.org
#     False Positive reports:    fp@sourcefire.com
#     Snort bugs:                bugs@snort.org
#
#     Compatible with Snort Versions:
#     VERSIONS : 2.9.15.1
#
#     Snort build options:
#     OPTIONS : --enable-gre --enable-mpls --enable-targetbased --enable-ppm --enable-perfprofi>
#
#     Additional information:
#     This configuration file enables active response, to run snort in
#     test mode -T you are required to supply an interface -i <interface>
#     or test mode will fail to fully validate the configuration and
#     exit with a FATAL error
#--------------------------------------------------

###################################################
# This file contains a sample snort configuration.
# You should take the following steps to create your own custom configuration:
#
                              [ Read 756 lines ]
^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line
```

Step 4: Update and Manage Snort Rules

1. Check out the various rule files in the rules directory. Which rules stick out to you? What is the purpose of rules in general?

Rules that stick out to me are the scan rules because we just finished our port scanning unit. This rule can detect if a port is being scanned by an outside entity. The purpose of Snort rules are to customize the boundaries and security that a user wishes to set. This makes the defense of the network more adaptable to what a user needs.

Step 5: Test Snort Configuration

```
Snort successfully validated the configuration!
Snort exiting
kirbical@kyraserver:~$
```

Step 6: Running Snort in IDS Mode

```
kirbical@kyraserver:~$ sudo snort -c /etc/snort/snort.conf -i eth0
Running in IDS mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
PortVar 'HTTP_PORTS' defined :  [ 80:81 311 383 591 593 901 1220 1414 1741 1830
2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:7145 7510 7777
7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 8300
8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002
55555 ]
PortVar 'SHELLCODE_PORTS' defined :  [ 0:79 81:65535 ]
PortVar 'ORACLE_PORTS' defined :  [ 1024:65535 ]
PortVar 'SSH_PORTS' defined :  [ 22 ]
PortVar 'FTP_PORTS' defined :  [ 21 2100 3535 ]
PortVar 'SIP_PORTS' defined :  [ 5060:5061 5600 ]
PortVar 'FILE_DATA_PORTS' defined :  [ 80:81 110 143 311 383 591 593 901 1220 14
14 1741 1830 2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:71
45 7510 7777 7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 82
43 8280 8300 8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444
 41080 50002 55555 ]
PortVar 'GTP_PORTS' defined :  [ 2123 2152 3386 ]
Detection:
   Search-Method = AC-Full-Q
```

Step 7: Viewing Snort Logs

2. Go to the */var/log/snort/* directory. What files did you find here? Do any of them contain any content? Why or why not?

I found the following files in the directory. The files are empty because they have not detected any events yet. This is because the detection modes are not enabled.

```
kirbical@kyraserver:/var/log/snort$ ls
snort.alert.1.gz  snort.alert.fast       snort.alert.fast.3.gz
snort.alert.2.gz  snort.alert.fast.1.gz
snort.alert.3.gz  snort.alert.fast.2.gz
```

Step 8: Running Snort as a Daemon

```
kirbical@kyraserver:~$ sudo snort -D -c etc/snort/snort.conf -i -ens160
top - 16:42:49 up 22 min,  1 user,  load average: 0.10, 0.09, 0.12
Tasks: 284 total,   1 running, 283 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.7 us,  0.3 sy,  0.0 ni, 99.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   3908.8 total,    656.9 free,    864.7 used,   2387.2 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.   2818.8 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   2018 kirbical  20   0 4277292 381396 128308 S   0.7   9.5   1:05.74 gnome-shell
  19699 root      20   0  167688   8028   6696 S   0.3   0.2   0:01.74 vmtoolsd
  33997 systemd+  20   0   15052   3804   3060 S   0.3   0.1   0:01.93 systemd-oomd
  34462 kirbical  20   0  561896  50608  38228 S   0.3   1.3   0:00.75 gnome-terminal-
  34535 kirbical  20   0   10112   3596   2744 R   0.3   0.1   0:00.14 top
      1 root      20   0  249416  11744   7644 S   0.0   0.3   0:03.95 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_hig+
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_rude_
     12 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_trace
     13 root      20   0       0      0      0 S   0.0   0.0   0:00.21 ksoftirqd/0
     14 root      20   0       0      0      0 I   0.0   0.0   0:00.58 rcu_sched
     15 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 migration/0
     16 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
     17 root      20   0       0      0      0 I   0.0   0.0   0:00.13 kworker/0:1-cgroup_dest+
     18 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
     19 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
     20 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/1
     21 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 migration/1
```

3. If you wanted to stop the Snort process from running, what is the command to terminate it?

If you wanted to terminate Snort you would use the “kill” command.