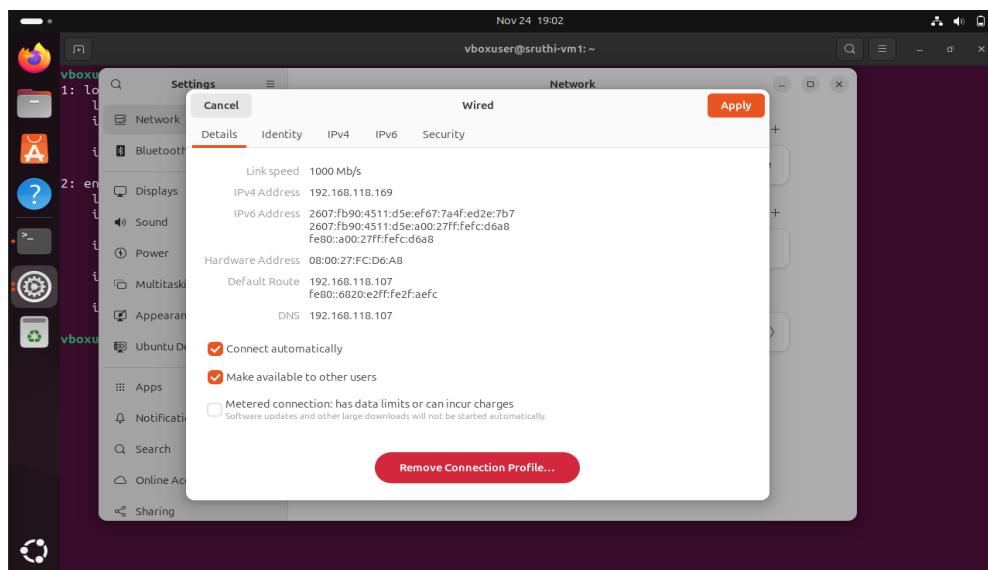


PART1:

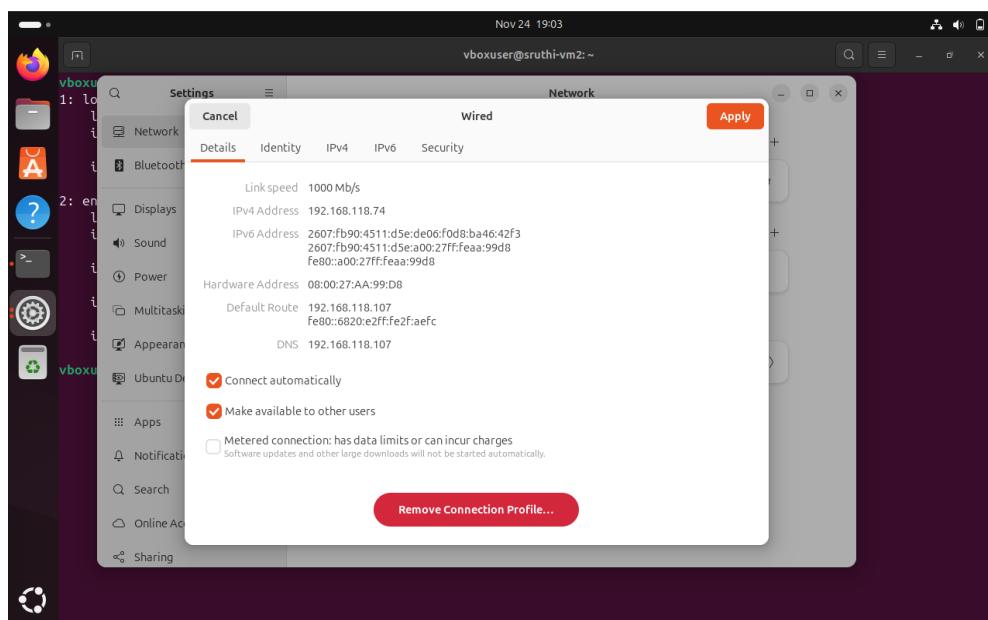
A: Configure Virtual Machines (VMs)



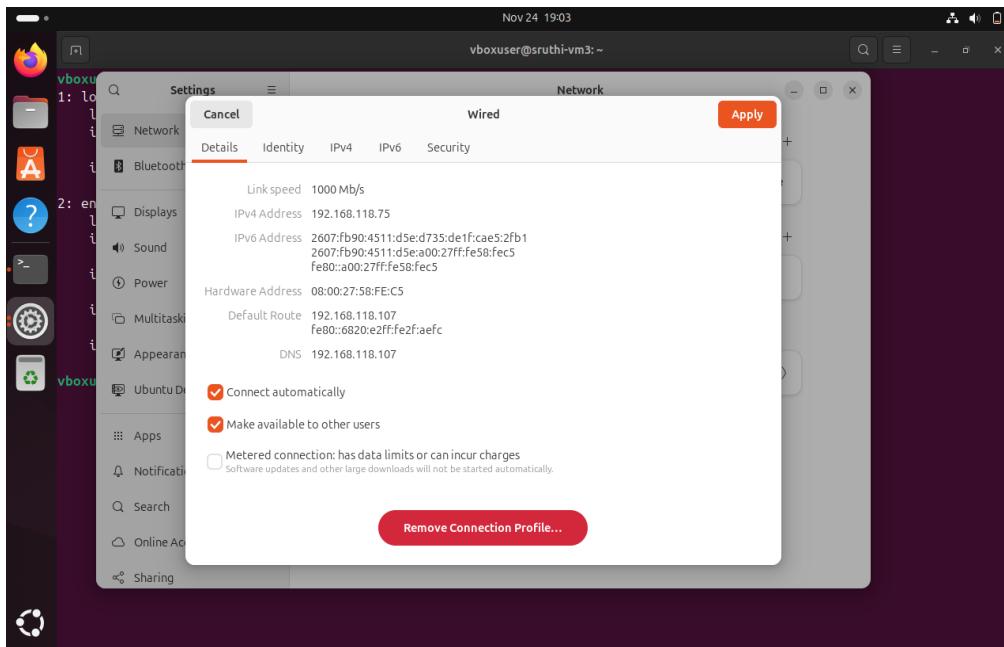
VM1 ip-address:



VM2 ip-address:



VM3 ip-address:



Ping requests – vm1

```
Nov 24 19:22
vboxuser@sruhti-vm1:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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 noprefixroute
            valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:58:fe:c5 brd ff:ff:ff:ff:ff:ff
        inet 192.168.118.118/24 brd 192.168.118.255 scope global dynamic noprefixroute enp0s3
            valid_lft 3581sec preferred_lft 3581sec
        inet6 2607:fb90:4511:d5e:d735:de1f:cae5:2fb5
            valid_lft 7447sec preferred_lft 7447sec
        inet6 2607:fb90:4511:d5e:a00:27ff:fe2f:aefc
            valid_lft 7182sec preferred_lft 7182sec
        inet6 fe80::a00:27ff:fe2f:aefc/64 scope link
            valid_lft forever preferred_lft forever
vboxuser@sruhti-vm1:~$ ping 192.168.118.74
PING 192.168.118.74 (192.168.118.74) 56(84) bytes of data.
64 bytes from 192.168.118.74: icmp_seq=1 ttl=64 time=6.09 ms
64 bytes from 192.168.118.74: icmp_seq=2 ttl=64 time=2.88 ms
64 bytes from 192.168.118.74: icmp_seq=3 ttl=64 time=3.00 ms
64 bytes from 192.168.118.74: icmp_seq=4 ttl=64 time=0.987 ms
64 bytes from 192.168.118.74: icmp_seq=5 ttl=64 time=2.40 ms
64 bytes from 192.168.118.74: icmp_seq=6 ttl=64 time=39.9 ms
64 bytes from 192.168.118.74: icmp_seq=7 ttl=64 time=4.76 ms
64 bytes from 192.168.118.74: icmp_seq=8 ttl=64 time=10.9 ms
64 bytes from 192.168.118.74: icmp_seq=9 ttl=64 time=1.70 ms
64 bytes from 192.168.118.74: icmp_seq=10 ttl=64 time=3.45 ms
64 bytes from 192.168.118.74: icmp_seq=11 ttl=64 time=2.49 ms
64 bytes from 192.168.118.74: icmp_seq=12 ttl=64 time=2.38 ms
64 bytes from 192.168.118.74: icmp_seq=13 ttl=64 time=1.31 ms
64 bytes from 192.168.118.74: icmp_seq=14 ttl=64 time=1.02 ms
```

```
Nov 24 19:23
vboxuser@sruhti-vm1:~$ ping 192.168.118.75
PING 192.168.118.75 (192.168.118.75) 56(84) bytes of data.
64 bytes from 192.168.118.75: icmp_seq=1 ttl=64 time=12.4 ms
64 bytes from 192.168.118.75: icmp_seq=2 ttl=64 time=1.59 ms
64 bytes from 192.168.118.75: icmp_seq=3 ttl=64 time=1.93 ms
64 bytes from 192.168.118.75: icmp_seq=4 ttl=64 time=2.25 ms
64 bytes from 192.168.118.75: icmp_seq=5 ttl=64 time=0.893 ms
64 bytes from 192.168.118.75: icmp_seq=6 ttl=64 time=2.45 ms
64 bytes from 192.168.118.75: icmp_seq=7 ttl=64 time=1.40 ms
64 bytes from 192.168.118.75: icmp_seq=8 ttl=64 time=2.68 ms
64 bytes from 192.168.118.75: icmp_seq=9 ttl=64 time=1.47 ms
^C
--- 192.168.118.75 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8033ms
rtt min/avg/max/mdev = 0.893/3.007/12.405/3.364 ms
vboxuser@sruhti-vm1:~$
```

Ping requests - vm2:

```
Nov 24 19:24
vboxuser@sruthi-vm2:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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 noprefixroute
            valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:00:27:aa:99:d8 brd ff:ff:ffff:ff:ff:ff
        inet 192.168.118.74/24 brd 192.168.118.255 scope global dynamic noprefixroute enp0s3
            valid_lft 3315sec preferred_lft 3315sec
        inet6 2607:fb90:4511:d5e:127:bc7e:8ca2:549e/64 scope global temporary dynamic
            valid_lft 7180sec preferred_lft 7180sec
        inet6 2607:fb90:4511:d5e:a00:27ff:feaa:99d8/64 scope global dynamic mngtmpaddr
            valid_lft 7180sec preferred_lft 7180sec
        inet6 fe80::a00:27ff:feaa:99d8/64 scope link
            valid_lft forever preferred_lft forever
vboxuser@sruthi-vm2:~$ ping 192.168.118.169
PING 192.168.118.169 (192.168.118.169) 56(84) bytes of data.
64 bytes from 192.168.118.169: icmp_seq=1 ttl=64 time=2.47 ms
64 bytes from 192.168.118.169: icmp_seq=2 ttl=64 time=1.01 ms
64 bytes from 192.168.118.169: icmp_seq=3 ttl=64 time=2.50 ms
64 bytes from 192.168.118.169: icmp_seq=4 ttl=64 time=1.02 ms
64 bytes from 192.168.118.169: icmp_seq=5 ttl=64 time=1.31 ms
64 bytes from 192.168.118.169: icmp_seq=6 ttl=64 time=2.65 ms
64 bytes from 192.168.118.169: icmp_seq=7 ttl=64 time=1.78 ms
64 bytes from 192.168.118.169: icmp_seq=8 ttl=64 time=2.59 ms
64 bytes from 192.168.118.169: icmp_seq=9 ttl=64 time=0.966 ms
64 bytes from 192.168.118.169: icmp_seq=10 ttl=64 time=1.31 ms
64 bytes from 192.168.118.169: icmp_seq=11 ttl=64 time=1.97 ms
64 bytes from 192.168.118.169: icmp_seq=12 ttl=64 time=2.29 ms
64 bytes from 192.168.118.169: icmp_seq=13 ttl=64 time=1.84 ms
64 bytes from 192.168.118.169: icmp_seq=14 ttl=64 time=1.05 ms
^C
```

```
Nov 24 19:24
vboxuser@sruthi-vm2:~$ ^C
--- 192.168.118.169 ping statistics ---
39 packets transmitted, 39 received, 0% packet loss, time 38073ms
rtt min/avg/max/mdev = 0.957/1.779/3.288/0.635 ms
vboxuser@sruthi-vm2:~$ ping 192.168.118.75
PING 192.168.118.75 (192.168.118.75) 56(84) bytes of data.
64 bytes from 192.168.118.75: icmp_seq=1 ttl=64 time=4.40 ms
64 bytes from 192.168.118.75: icmp_seq=2 ttl=64 time=1.59 ms
64 bytes from 192.168.118.75: icmp_seq=3 ttl=64 time=2.91 ms
64 bytes from 192.168.118.75: icmp_seq=4 ttl=64 time=1.27 ms
64 bytes from 192.168.118.75: icmp_seq=5 ttl=64 time=2.94 ms
64 bytes from 192.168.118.75: icmp_seq=6 ttl=64 time=1.57 ms
64 bytes from 192.168.118.75: icmp_seq=7 ttl=64 time=1.13 ms
64 bytes from 192.168.118.75: icmp_seq=8 ttl=64 time=1.23 ms
64 bytes from 192.168.118.75: icmp_seq=9 ttl=64 time=1.17 ms
64 bytes from 192.168.118.75: icmp_seq=10 ttl=64 time=0.914 ms
64 bytes from 192.168.118.75: icmp_seq=11 ttl=64 time=0.965 ms
64 bytes from 192.168.118.75: icmp_seq=12 ttl=64 time=1.05 ms
64 bytes from 192.168.118.75: icmp_seq=13 ttl=64 time=1.42 ms
64 bytes from 192.168.118.75: icmp_seq=14 ttl=64 time=2.03 ms
64 bytes from 192.168.118.75: icmp_seq=15 ttl=64 time=1.99 ms
64 bytes from 192.168.118.75: icmp_seq=16 ttl=64 time=1.97 ms
64 bytes from 192.168.118.75: icmp_seq=17 ttl=64 time=3.19 ms
64 bytes from 192.168.118.75: icmp_seq=18 ttl=64 time=1.38 ms
64 bytes from 192.168.118.75: icmp_seq=19 ttl=64 time=1.79 ms
64 bytes from 192.168.118.75: icmp_seq=20 ttl=64 time=1.49 ms
64 bytes from 192.168.118.75: icmp_seq=21 ttl=64 time=2.88 ms
^C
--- 192.168.118.75 ping statistics ---
21 packets transmitted, 21 received, 0% packet loss, time 20047ms
rtt min/avg/max/mdev = 0.914/1.870/4.400/0.884 ms
vboxuser@sruthi-vm2:~$
```

Ping requests - vm3:

```
Nov 24 19:29
vboxuser@sruthi-vm3:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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 noprefixroute
            valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 00:00:27:58:fe:c5 brd ff:ff:ffff:ff:ff:ff
        inet 192.168.118.75/24 brd 192.168.118.255 scope global dynamic noprefixroute enp0s3
            valid_lft 3391sec preferred_lft 3391sec
        inet6 2607:fb90:4511:d5e:d735:de1f:cae5:2fb1/64 scope global temporary dynamic
            valid_lft 6993sec preferred_lft 6993sec
        inet6 2607:fb90:4511:d5e:a00:27ff:fe58:fec5/64 scope global dynamic mngtmpaddr
            valid_lft 6993sec preferred_lft 6993sec
        inet6 fe80::a00:27ff:fe58:fec5/64 scope link
            valid_lft forever preferred_lft forever
vboxuser@sruthi-vm3:~$ ping 192.168.118.74
PING 192.168.118.74 (192.168.118.74) 56(84) bytes of data.
64 bytes from 192.168.118.74: icmp_seq=1 ttl=64 time=2.81 ms
64 bytes from 192.168.118.74: icmp_seq=2 ttl=64 time=3.42 ms
64 bytes from 192.168.118.74: icmp_seq=3 ttl=64 time=2.33 ms
64 bytes from 192.168.118.74: icmp_seq=4 ttl=64 time=2.02 ms
64 bytes from 192.168.118.74: icmp_seq=5 ttl=64 time=2.19 ms
64 bytes from 192.168.118.74: icmp_seq=6 ttl=64 time=2.95 ms
64 bytes from 192.168.118.74: icmp_seq=7 ttl=64 time=1.79 ms
64 bytes from 192.168.118.74: icmp_seq=8 ttl=64 time=1.89 ms
64 bytes from 192.168.118.74: icmp_seq=9 ttl=64 time=1.95 ms
^C
--- 192.168.118.74 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8058ms
rtt min/avg/max/mdev = 1.789/2.372/3.423/0.531 ms
vboxuser@sruthi-vm3:~$
```

```

Nov 24 19:30
vboxuser@sruthi-vm3:~ valid_lft forever preferred_lft forever
vboxuser@sruthi-vm3:~ $ ping 192.168.118.74
PING 192.168.118.74 (192.168.118.74) 56(84) bytes of data.
64 bytes from 192.168.118.74: icmp_seq=1 ttl=64 time=2.81 ms
64 bytes from 192.168.118.74: icmp_seq=2 ttl=64 time=3.42 ms
64 bytes from 192.168.118.74: icmp_seq=3 ttl=64 time=2.33 ms
64 bytes from 192.168.118.74: icmp_seq=4 ttl=64 time=2.02 ms
64 bytes from 192.168.118.74: icmp_seq=5 ttl=64 time=2.19 ms
64 bytes from 192.168.118.74: icmp_seq=6 ttl=64 time=2.95 ms
64 bytes from 192.168.118.74: icmp_seq=7 ttl=64 time=1.79 ms
64 bytes from 192.168.118.74: icmp_seq=8 ttl=64 time=1.89 ms
64 bytes from 192.168.118.74: icmp_seq=9 ttl=64 time=1.95 ms
^C
--- 192.168.118.74 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8058ms
rtt min/avg/max/mdev = 1.789/2.372/3.423/0.531 ms
vboxuser@sruthi-vm3:~ $ ping 192.168.118.169
PING 192.168.118.169 (192.168.118.169) 56(84) bytes of data.
64 bytes from 192.168.118.169: icmp_seq=1 ttl=64 time=1.62 ms
64 bytes from 192.168.118.169: icmp_seq=2 ttl=64 time=1.44 ms
64 bytes from 192.168.118.169: icmp_seq=3 ttl=64 time=2.00 ms
64 bytes from 192.168.118.169: icmp_seq=4 ttl=64 time=2.51 ms
64 bytes from 192.168.118.169: icmp_seq=5 ttl=64 time=1.72 ms
64 bytes from 192.168.118.169: icmp_seq=6 ttl=64 time=2.28 ms
64 bytes from 192.168.118.169: icmp_seq=7 ttl=64 time=1.56 ms
64 bytes from 192.168.118.169: icmp_seq=8 ttl=64 time=1.91 ms
64 bytes from 192.168.118.169: icmp_seq=9 ttl=64 time=1.53 ms
^C
--- 192.168.118.169 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8677ms
rtt min/avg/max/mdev = 1.437/1.839/2.508/0.345 ms
Show Apps sruthi-vm3:~ $ 

```

Each VM Internet Access:

VM1:

```

[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 07:03:32]
$ ping -c 4 google.com
PING google.com (2607:f8b0:4023:1002::64) 56 data bytes

--- google.com ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3103ms

[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 07:04:26]
$ 

```

VM2:

```

Nov 25 07:00
Terminal
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

[sruthi-mandalapu vboxuser sruthi-vm2 ~ 2024-11-25 06:56:20]
$ ping -c 4 google.com
PING google.com (2607:f8b0:4023:1002::8b) 56 data bytes

--- google.com ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3059ms

[sruthi-mandalapu vboxuser sruthi-vm2 ~ 2024-11-25 07:00:09]
$ 

```

VM3:

```

$ ping -c 4 google.com
PING google.com (2607:f8b0:4023:1002::8b) 56 data bytes

--- google.com ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3088ms

[sruthi-mandalapu vboxuser sruthi-vm3 ~ 2024-11-25 07:05:49]
$ 

```

B: Customize the bash prompt

VM1:

```

--- 192.168.118.74 ping statistics ---
118 packets transmitted, 118 received, 0% packet loss, time 119756ms
rtt min/avg/max/mdev = 0.658/2.974/39.885/4.455 ms
vboxuser@sruthi-vm1: $ ping 192.168.118.75
PING 192.168.118.75 (192.168.118.75) 56(84) bytes of data.
64 bytes from 192.168.118.75: icmp_seq=1 ttl=64 time=12.4 ms
64 bytes from 192.168.118.75: icmp_seq=2 ttl=64 time=1.59 ms
64 bytes from 192.168.118.75: icmp_seq=3 ttl=64 time=1.93 ms
64 bytes from 192.168.118.75: icmp_seq=4 ttl=64 time=2.25 ms
64 bytes from 192.168.118.75: icmp_seq=5 ttl=64 time=0.893 ms
64 bytes from 192.168.118.75: icmp_seq=6 ttl=64 time=2.45 ms
64 bytes from 192.168.118.75: icmp_seq=7 ttl=64 time=1.40 ms
64 bytes from 192.168.118.75: icmp_seq=8 ttl=64 time=2.68 ms
64 bytes from 192.168.118.75: icmp_seq=9 ttl=64 time=1.47 ms
^C
--- 192.168.118.75 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8033ms
rtt min/avg/max/mdev = 0.893/3.007/12.405/3.364 ms
vboxuser@sruthi-vm1: $ nano ~/.bashrc

```

```

GNU nano 7.2
/home/vboxuser/.bashrc: ~

alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands. Use like so:
# sleep 10; alert
alias alert='notify-send --urgency=low -i "$(($? == 0) && echo terminal || echo error)" "$(history|tail -n1|sed -e '\''s/^\!.*$/'\'')"'"

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
  . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc)
if ! shopt -o posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
PS1='[sruthi-mandalapu ~]$ \w \h \w $(date "+%Y-%m-%d %H:%M:%S")] \w\$'
File Name to Write: /home/vboxuser/.bashrc
^C Help           M-D DOS Format      M-A Append
^C Cancel         M-M Mac Format      M-P Prepend
^M-B Backup File
^T Browse

```

```

64 bytes from 192.168.118.75: icmp_seq=5 ttl=64 time=0.893 ms
64 bytes from 192.168.118.75: icmp_seq=6 ttl=64 time=2.45 ms
64 bytes from 192.168.118.75: icmp_seq=7 ttl=64 time=1.40 ms
64 bytes from 192.168.118.75: icmp_seq=8 ttl=64 time=2.68 ms
64 bytes from 192.168.118.75: icmp_seq=9 ttl=64 time=1.47 ms
^C
--- 192.168.118.75 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8033ms
rtt min/avg/max/mdev = 0.893/3.007/12.405/3.364 ms
vboxuser@sruthi-vm1: $ nano ~/.bashrc
vboxuser@sruthi-vm1: $ source ~/.bashrc
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-24 19:49:30]
$ 

```

VM2:

```

64 bytes from 192.168.118.75: icmp_seq=13 ttl=64 time=1.42 ms
64 bytes from 192.168.118.75: icmp_seq=14 ttl=64 time=2.03 ms
64 bytes from 192.168.118.75: icmp_seq=15 ttl=64 time=1.99 ms
64 bytes from 192.168.118.75: icmp_seq=16 ttl=64 time=1.97 ms
64 bytes from 192.168.118.75: icmp_seq=17 ttl=64 time=3.19 ms
64 bytes from 192.168.118.75: icmp_seq=18 ttl=64 time=1.38 ms
64 bytes from 192.168.118.75: icmp_seq=19 ttl=64 time=1.79 ms
64 bytes from 192.168.118.75: icmp_seq=20 ttl=64 time=1.49 ms
64 bytes from 192.168.118.75: icmp_seq=21 ttl=64 time=2.88 ms
^C
--- 192.168.118.75 ping statistics ---
21 packets transmitted, 21 received, 0% packet loss, time 20047ms
rtt min/avg/max/mdev = 0.914/1.870/4.400/0.884 ms
vboxuser@sruthi-vm2: $ nano ~/.bashrc

```

```

Nov 24 20:04
vboxuser@sruthi-vm2:~ /home/vboxuser/.bashrc *

GNU nano 7.2
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands. Use like so:
# sleep 10; alert
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)" "$(history|tail -n1|sed -e '\''s/
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
. ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -o posix; then
if [ -f /usr/share/bash-completion/bash_completion ]; then
. /usr/share/bash-completion/bash_completion
elif [ -f /etc/bash_completion ]; then
. /etc/bash_completion
fi
fi
PS1='[sruthi-mandalapu \u \h \w $(date "+%Y-%m-%d %H:%M:%S")] \n$ '

^G Help      ^O Write Out   ^W Where Is    ^K Cut        ^T Execute   ^C Location   M-U Undo
^X Exit      ^R Read File   ^A Replace     ^U Paste     ^J Justify   ^Y Go To Line M-E Redo
                                         M-A Set Mark M-6 Copy

```

vboxuser@sruthi-vm2:~\$ source ~/.bashrc
vboxuser@sruthi-vm2:~\$ nano ~/.bashrc
vboxuser@sruthi-vm2:~\$ source ~/.bashrc
[sruthi-mandalapu vboxuser sruthi-vm2 ~ 2024-11-24 20:04:46]

VM3:

```

9 packets transmitted, 9 received, 0% packet loss, time 8677ms
rtt min/avg/max/mdev = 1.437/1.839/2.508/0.345 ms
vboxuser@sruthi-vm3:~$ source ~/.bashrc
vboxuser@sruthi-vm3:~$ nano ~/.bashrc
vboxuser@sruthi-vm3:~$ source ~/.bashrc
vboxuser@sruthi-vm3:~$ 

```

```

Nov 24 20:14
vboxuser@sruthi-vm3:~ /home/vboxuser/.bashrc

GNU nano 7.2
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands. Use like so:
# sleep 10; alert
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)" "$(history|tail -n1|sed -e '\''s/
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
. ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -o posix; then
if [ -f /usr/share/bash-completion/bash_completion ]; then
. /usr/share/bash-completion/bash_completion
elif [ -f /etc/bash_completion ]; then
. /etc/bash_completion
fi
fi
PS1='[sruthi-mandalapu \u \h \w $(date "+%Y-%m-%d %H:%M:%S")] \n$ '

^G Help      ^O Write Out   ^W Where Is    ^K Cut        ^T Execute   ^C Location   M-U Undo
^X Show Apps ^R Read File   ^A Replace     ^U Paste     ^J Justify   ^Y Go To Line M-E Redo
                                         M-A Set Mark M-6 Copy

rtt min/avg/max/mdev = 1.437/1.839/2.508/0.345 ms
vboxuser@sruthi-vm3:~$ source ~/.bashrc
vboxuser@sruthi-vm3:~$ nano ~/.bashrc
vboxuser@sruthi-vm3:~$ nano ~/.bashrc
vboxuser@sruthi-vm3:~$ source ~/.bashrc
[sruthi-mandalapu vboxuser sruthi-vm3 ~ 2024-11-24 20:15:32]
$ 

```

PART2: Baseline Analysis (Before SYN Flood Attack)

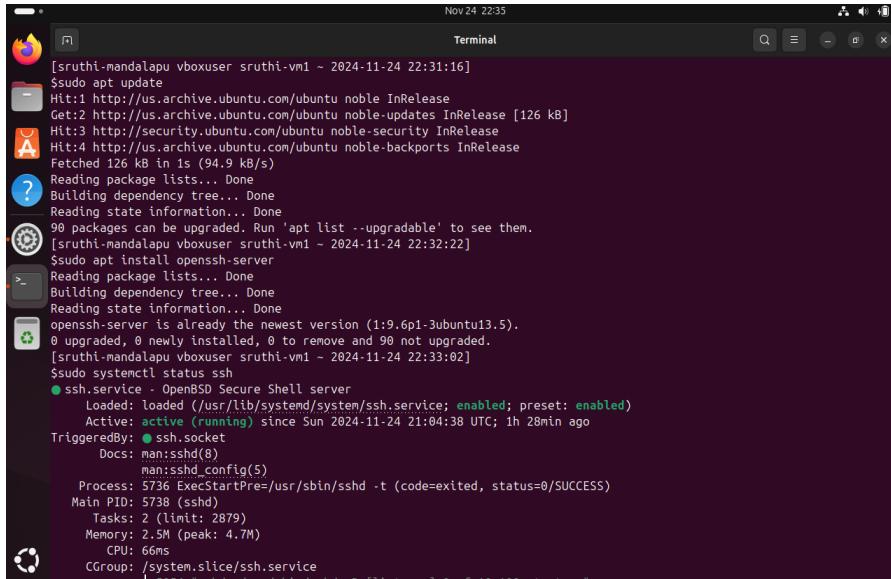
a)

Enable SSH on VM1: commands

sudo apt update

sudo apt install openssh-server

`sudo systemctl status ssh`



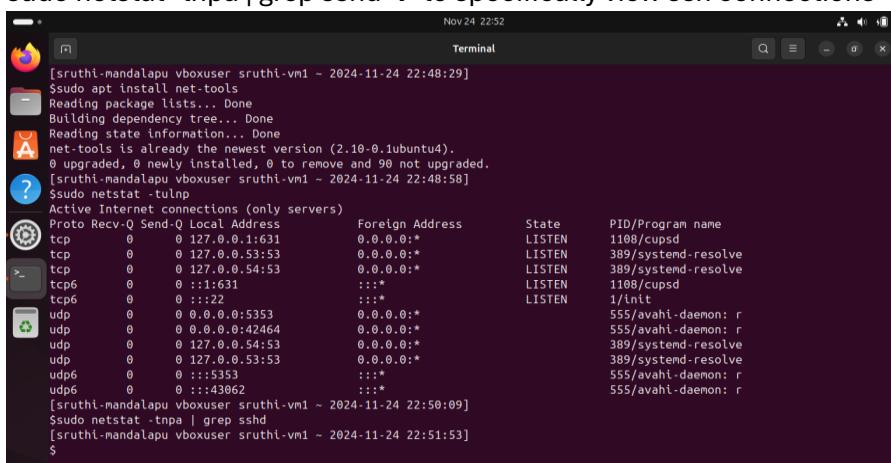
```
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-24 22:31:16]
$ sudo apt update
Hit:1 http://us.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu noble-backports InRelease
Fetched 126 kB in 1s (94.9 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
98 packages can be upgraded. Run 'apt list --upgradable' to see them.
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-24 22:32:22]
$ sudo apt install openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:9.6p1-3ubuntu13.5).
0 upgraded, 0 newly installed, 0 to remove and 90 not upgraded.
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-24 22:33:02]
$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
     Active: active (running) since Sun 2024-11-24 21:04:38 UTC; 1h 28min ago
TriggeredBy: ● ssh.socket
   Docs: man:sshd(8)
          man:sshd_config(5)
    Process: 5736 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 5738 (sshd)
      Tasks: 2 (limit: 2879)
     Memory: 2.5M (peak: 4.7M)
        CPU: 66ms
       CGroup: /system.slice/ssh.service
           └─ 5738 sshd[5738] /usr/sbin/sshd -D listener[0 of 10-100 startups]
```

Initial State on VM1: commands

`sudo apt install net-tools`

`sudo netstat -tulnp` → view active network connections

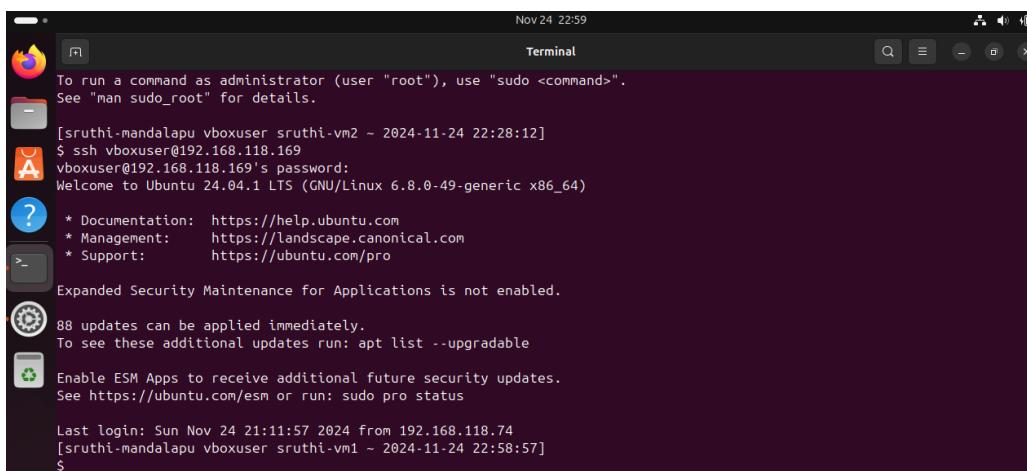
`sudo netstat -tnpa | grep sshd` → to specifically view ssh connections



```
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-24 22:48:29]
$ sudo apt install net-tools
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
net-tools is already the newest version (2.10-0.1ubuntu4).
0 upgraded, 0 newly installed, 0 to remove and 90 not upgraded.
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-24 22:48:58]
$ sudo netstat -tulnp
Active Internet connections (only servers)
Proto Recv-Q Local Address          Foreign Address        State      PID/Program name
tcp     0      0.0.0.0:1631          0.0.0.0:*          LISTEN    1108/cupsd
tcp     0      0.0.0.0:53:53         0.0.0.0:*          LISTEN    389/systemd-resolve
tcp     0      0.0.0.0:54:53         0.0.0.0:*          LISTEN    389/systemd-resolve
tcp6    0      0::1:631             ::*:*                LISTEN    1108/cupsd
tcp6    0      0::2:2              ::*:*                LISTEN    1/init
tcp6    0      0.0.0.0:5353        0.0.0.0:*          LISTEN    555/avahi-daemon: r
udp     0      0.0.0.0:42464       0.0.0.0:*          LISTEN    555/avahi-daemon: r
udp     0      0.0.0.0:54:53         0.0.0.0:*          LISTEN    389/systemd-resolve
udp     0      0.0.0.0:53:53         0.0.0.0:*          LISTEN    389/systemd-resolve
udp6    0      0:::5353            ::*:*                LISTEN    555/avahi-daemon: r
udp6    0      0:::43062           ::*:*                LISTEN    555/avahi-daemon: r
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-24 22:50:09]
$ sudo netstat -tnpa | grep sshd
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-24 22:51:53]
$
```

Use VM2 to Establish SSH connection with VM1: commands

`ssh username_of_VM1@VM1_IPAddress`



```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

[sruthi-mandalapu vboxuser sruthi-vm2 ~ 2024-11-24 22:28:12]
$ ssh vboxuser@192.168.118.169
vboxuser@192.168.118.169's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-49-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

88 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sun Nov 24 21:11:57 2024 from 192.168.118.74
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-24 22:58:57]
$
```

Execute series of commands during SSH session: commands

`ls, pwd, date`

```
Building dependency tree... Done
Reading state information... Done
net-tools is already the newest version (2.10-0.1ubuntu4).
0 upgraded, 0 newly installed, 0 to remove and 90 not upgraded.
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-24 22:48:58]
Ssudo netstat -tulpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State      PID/Program name
tcp        0      0 127.0.0.1:631          0.0.0.0:*              LISTEN     1108/cupsd
tcp        0      0 127.0.0.53:53          0.0.0.0:*              LISTEN     389/systemd-resolve
tcp        0      0 127.0.0.54:53          0.0.0.0:*              LISTEN     389/systemd-resolve
tcp6       0      0 ::1:631               ::*:*                  LISTEN     1108/cupsd
tcp6       0      0 ::1:22                ::*:*                  LISTEN     1/init
udp        0      0 0.0.0.0:5353          0.0.0.0:*              LISTEN     555/avahi-daemon: r
udp        0      0 0.0.0.0:42464         0.0.0.0:*              LISTEN     555/avahi-daemon: r
udp        0      0 127.0.0.54:53          0.0.0.0:*              LISTEN     389/systemd-resolve
udp        0      0 127.0.0.53:53          0.0.0.0:*              LISTEN     389/systemd-resolve
udp6       0      0 ::1:5353             ::*:*                  LISTEN     555/avahi-daemon: r
udp6       0      0 ::1:43062            ::*:*                  LISTEN     555/avahi-daemon: r
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-24 22:50:09]
Ssudo netstat -tnpa | grep sshd
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-24 22:51:53]
$ls
Desktop Documents Downloads Music Pictures Public snap Templates Videos
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-24 23:09:13]
$pwd
/home/vboxuser
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-24 23:09:16]
$date
Sun Nov 24 11:09:18 PM UTC 2024
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-24 23:09:18]
```

Pre-Attack State on VM1: commands

`sudo netstat -tulnp` → view active network connections

`sudo netstat -tnpa | grep sshd` → to specifically view ssh connections

```
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-24 22:50:09]
$ sudo netstat -tnpa | grep sshd
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-24 22:51:53]
$ ls
Desktop Documents Downloads Music Pictures Public snap Templates Videos
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-24 23:09:13]
$ pwd
/home/vboxuser
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-24 23:09:16]
$ date
Sun Nov 24 11:09:18 PM UTC 2024
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-24 23:09:18]
$ sudo netstat -tulpn
[sudo] password for vboxuser:
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp        0      0 127.0.0.1:631           0.0.0.0:*          LISTEN    1108/cupsd
tcp        0      0 127.0.0.53:53          0.0.0.0:*          LISTEN    389/systemd-resolve
tcp        0      0 127.0.0.54:53          0.0.0.0:*          LISTEN    389/systemd-resolve
tcp6       0      0 ::1:631              ::*:*               LISTEN    1108/cupsd
tcp6       0      0 ::1:22               ::*:*               LISTEN    1/init
udp        0      0 0.0.0.0:5353          0.0.0.0:*          LISTEN    555/avahi-daemon: r
udp        0      0 0.0.0.0:42464         0.0.0.0:*          LISTEN    555/avahi-daemon: r
udp        0      0 127.0.0.54:53          0.0.0.0:*          LISTEN    389/systemd-resolve
udp        0      0 127.0.0.53:53          0.0.0.0:*          LISTEN    389/systemd-resolve
udp6       0      0 ::5353              ::*:*               LISTEN    555/avahi-daemon: r
udp6       0      0 ::43062             ::*:*               LISTEN    555/avahi-daemon: r
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-24 23:15:25]
$ sudo netstat -tnpa | grep sshd
tcp6       0      0 192.168.118.169:22     192.168.118.74:50600   ESTABLISHED 7153/ssh: vboxuser
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-24 23:16:08]
$
```

As we can observe there is a difference in initial state and pre-attack state for ssh connection. For initial state there are no ssh connection, if we observe there is a ssh connection established in pre-attack state. And the connection is between vm1 and vm2. We can find their ip addresses.

b)

Install wireshark:

```

Nov 25 01:05
Terminal

** (wireshark:9560) 00:26:10.850001 [Capture MESSAGE] -- Capture Start ...
** (wireshark:9560) 00:26:10.914768 [Capture MESSAGE] -- Error message from child: "Couldn't run dumpcap in child process: Permission denied", "(null)"
** (wireshark:9560) 00:26:34.691197 [Capture MESSAGE] -- Capture stopped.
** (wireshark:9560) 00:26:34.691246 [Capture WARNING] ./ui/capture.c:722 -- capture_input_closed():
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 00:26:36]
Sgetenv group wireshark
wireshark:x:124:
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 00:26:53]
$sudo usermod -aG wireshark $(whoami)
[sudo] password for vboxuser:
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 00:28:08]
$sudo usermod -aG wireshark $(whoami)
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 00:28:15]
$sudo wireshark
** (wireshark:9766) 00:28:51.694935 [GUI WARNING] -- QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
** (wireshark:9766) 00:29:20.196007 [Capture MESSAGE] -- Capture Start ...
** (wireshark:9766) 00:29:20.340773 [Capture MESSAGE] -- Capture started
** (wireshark:9766) 00:29:20.340874 [Capture MESSAGE] -- File: "/tmp/wireshark_enp0s32NANX2.pcapng"
** (wireshark:9766) 00:49:37.248046 [Capture MESSAGE] -- Capture Stop ...
** (wireshark:9766) 00:49:37.276578 [Capture MESSAGE] -- Capture stopped.
** (wireshark:9766) 00:49:37.276858 [Capture WARNING] ./ui/capture.c:722 -- capture_input_closed():
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 00:49:41]
$sudo wireshark
[sudo] password for vboxuser:
** (wireshark:10237) 00:50:40.231501 [GUI WARNING] -- QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
** (wireshark:10237) 00:50:45.945446 [Capture MESSAGE] -- Capture Start ...
** (wireshark:10237) 00:50:46.002796 [Capture MESSAGE] -- Capture started
** (wireshark:10237) 00:50:46.002886 [Capture MESSAGE] -- File: "/tmp/wireshark_enp0s3I3Q4X2.pcapng"

```

TCP Connection State: SYN_SENT

The main working of syn_sent is the client sends a syn (synchronize) packet to initiate a connection with the server. In wireshark look for SYN flag set in a packet with no ACK. It indicates that the client is attempting to establish a connection.

For SYN_SENT – the syn flag is set, but ack flag is not set which explains about syn_sent state. Syn sent is happens before getting acknowledgement from the receiver, hence ack is ‘0’ which is not set, syn is ‘1’ which is set. So, within Wireshark I have used filter command – “tcp.flags.syn==1 && tcp.flags.ack==0”. This gives the syn_sent requests – my VM1 ip-address is 192.168.118.169, my VM2 ip-address is 192.168.118.74. As you can observe the blue color highlighted things VM2 made a syn_sent to VM1, and also we could observe their respective source and destination ip-addresses.

| Time | Source | Destination | Protocol | Length | Info |
|--|---------------------|-------------|----------|----------------------|--|
| 73.122.340922546 2607:f9b0:4511:d5e.. | 2628:2d:4000:1::23 | TCP | 94 | 58364 -> 80 [SYN] | Seq=0 Win=64672 Len=0 MSS=1376 SACK_PERM |
| 74.123.393143398 2607:f9b0:4511:d5e.. | 2620:2d:4000:1::23 | TCP | 94 | [TCP Retransmission] | 50364 -> 80 [SYN] |
| 75.123.468604257 2607:f9b0:4511:d5e.. | 2620:2d:4000:1::23 | TCP | 94 | [TCP Retransmission] | 50364 -> 80 [SYN] |
| 77.126.53557474 2607:f9b0:4511:d5e.. | 2620:2d:4000:1::23 | TCP | 94 | [TCP Retransmission] | 50364 -> 80 [SYN] |
| 89.127.563213482 2607:f9b0:4511:d5e.. | 2620:2d:4000:1::23 | TCP | 94 | [TCP Retransmission] | 50364 -> 80 [SYN] |
| 89.128.000000748 2607:f9b0:4511:d5e.. | 2620:2d:4000:1::23 | TCP | 94 | [TCP Retransmission] | 50364 -> 80 [SYN] |
| 87.141.959795165 2607:f9b0:4511:d5e.. | 2620:2d:4000:1::23 | TCP | 94 | [TCP Retransmission] | 50364 -> 80 [SYN] |
| 90.141.95551465 2607:f9b0:4511:d5e.. | 2620:2d:4000:1::23 | TCP | 94 | [TCP Retransmission] | 50364 -> 80 [SYN] |
| 123.211.569963329 192.168.118.109 | 91.189.91.98 | TCP | 74 | 55188 -> 80 [SYN] | Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM |
| 215.423.502606324 192.168.118.109 | 192.168.118.109 | TCP | 74 | 47256 -> 80 [SYN] | Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM |
| 215.423.502606324 192.168.118.109 | 192.168.118.109 | TCP | 94 | [TCP Retransmission] | 50364 -> 80 [SYN] |
| 259.433.660001400 2607:f9b0:4511:d5e.. | 2620:2d:4002:1::190 | TCP | 94 | [TCP Retransmission] | 34278 -> 80 [SYN] |
| 261.434.691629081 2607:f9b0:4511:d5e.. | 2620:2d:4002:1::190 | TCP | 94 | [TCP Retransmission] | 34278 -> 80 [SYN] |
| 261.435.753189111 2607:f9b0:4511:d5e.. | 2620:2d:4002:1::190 | TCP | 94 | [TCP Retransmission] | 34278 -> 80 [SYN] |
| 265.439.095334495 2607:f9b0:4511:d5e.. | 2620:2d:4002:1::190 | TCP | 94 | [TCP Retransmission] | 34278 -> 80 [SYN] |

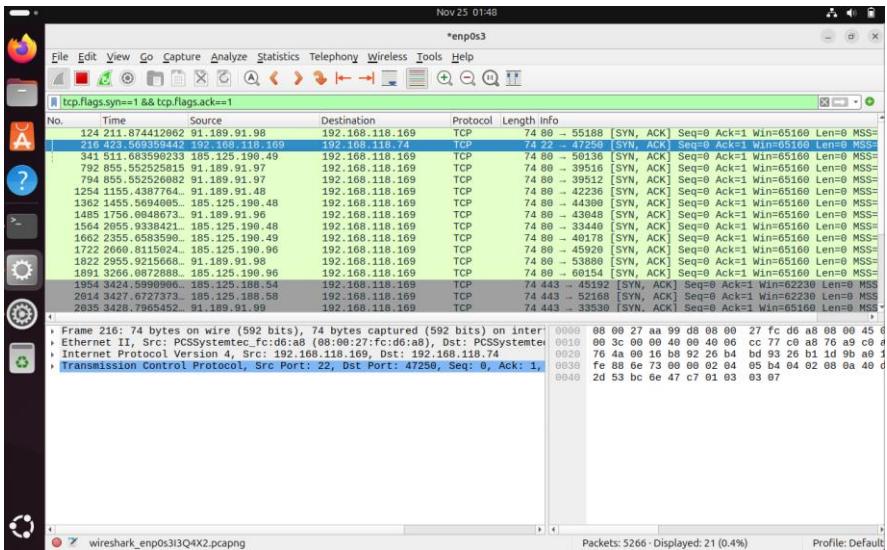
Packets: 613 - Displayed: 30 (4.9%) Profile: Default

TCP Connection State: SYN RECEIVED

The main working of syn_received is the server receives the syn packet and responds with a syn-ack packet. In wireshark look for a packet with both syn and ack flags are set. It shows the server is ready to establish the connection and awaits acknowledgement from the client.

For SYN_RECEIVED – the syn flag is set, but ack flag is set which explains about syn_received state. Syn received happens to give acknowledgement to the sender, it is generated from receiver, hence ack is ‘1’ which is set, syn is ‘1’ which is set. So, within Wireshark I have used filter command – “tcp.flags.syn==1 && tcp.flags.ack==1”. This gives the syn_received requests – my VM1 ip-address is 192.168.118.169, my VM2 ip-address is 192.168.118.74. As you can observe the blue color highlighted things VM1 made a syn_received acknowledgement to VM1, and also we could observe their respective source and destination ip-addresses. As,

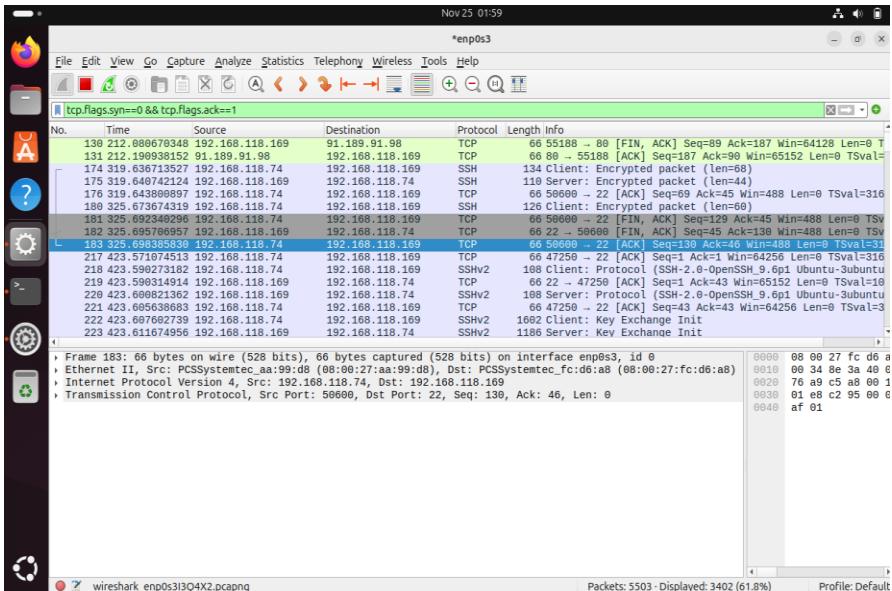
the acknowledgement need to be sent from vm1 – hence source ip-address is vm1, and vm2 as destination ip-addresses.



TCP Connection State: ESTABLISHED

The main working of established is the client acknowledges the server's SYN_ACK, completing the handshake. And then data transfer begins. In wireshark there is continuous flows of packets between client and server. This indicates an active connection.

For ESTABLISHED – the syn flag is not set, but ack flag is set which explains about established state. Established state happens when the connection is maintained between the source and destination, it is generated as ack is '1' which is set, syn is '0' which is not set. So, within Wireshark I have used filter command – "tcp.flags.syn==0 && tcp.flags.ack==1". This gives the connection established requests – my VM1 ip-address is 192.168.118.169, my VM2 ip-address is 192.168.118.74. As you can observe the blue color highlighted things VM2 made a established acknowledgement to VM1, and also we could observe their respective source and destination ip-addresses. As, the acknowledgement need to be sent from vm2 – hence source ip-address is vm2, and vm1 as destination ip-addresses.



PART3: Launching a SYN Flood Attack

a)

Command to check if cookies are enabled or not: `sysctl net.ipv4.tcp_syncookies`, as it is 1 – the cookies are enabled

```

Nov 25 02:47
Terminal

[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 00:26:36]
$ getent group wireshark
wireshark:x:124:
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 00:26:53]
$ sudo usermod -aG wireshark $whoami
[sudo] password for vboxuser:
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 00:28:08]
$ sudo usermod -aG wireshark $whoami
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 00:28:15]
$ sudo wireshark
** (wireshark:9766) 00:28:51.694935 [GUI WARNING] -- QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
** (wireshark:9766) 00:29:20.196007 [Capture MESSAGE] -- Capture Start ...
** (wireshark:9766) 00:29:20.340773 [Capture MESSAGE] -- Capture started
** (wireshark:9766) 00:29:20.340874 [Capture MESSAGE] -- File: "/tmp/wireshark_enp0s32NANX2.pcapng"
** (wireshark:9766) 00:49:37.248046 [Capture MESSAGE] -- Capture Stop ...
** (wireshark:9766) 00:49:37.276578 [Capture MESSAGE] -- Capture stopped.
** (wireshark:9766) 00:49:37.276858 [Capture WARNING] ./ui/capture.c:722 -- capture_input_closed():
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 00:49:41]
$ sudo wireshark
[sudo] password for vboxuser:
** (wireshark:10237) 00:50:40.231501 [GUI WARNING] -- QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
** (wireshark:10237) 00:50:45.945446 [Capture MESSAGE] -- Capture Start ...
** (wireshark:10237) 00:50:46.002706 [Capture MESSAGE] -- Capture started
** (wireshark:10237) 00:50:46.002886 [Capture MESSAGE] -- File: "/tmp/wireshark_enp0s3I3Q4X2.pcapng"
^C
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 02:46:12]
$ sysctl net.ipv4.tcp_syncookies
net.ipv4.tcp_syncookies = 1
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 02:47:15]
$ 
```

To disable cookies: sudo sysctl -w net.ipv4.tcp_syncookies=0

```

Nov 25 02:53
Terminal

$ sudo usermod -aG wireshark $whoami
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 00:28:15]
$ sudo wireshark
** (wireshark:9766) 00:28:51.694935 [GUI WARNING] -- QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
** (wireshark:9766) 00:29:20.196007 [Capture MESSAGE] -- Capture Start ...
** (wireshark:9766) 00:29:20.340773 [Capture MESSAGE] -- Capture started
** (wireshark:9766) 00:29:20.340874 [Capture MESSAGE] -- File: "/tmp/wireshark_enp0s32NANX2.pcapng"
** (wireshark:9766) 00:49:37.248046 [Capture MESSAGE] -- Capture Stop ...
** (wireshark:9766) 00:49:37.276578 [Capture MESSAGE] -- Capture stopped.
** (wireshark:9766) 00:49:37.276858 [Capture WARNING] ./ui/capture.c:722 -- capture_input_closed():
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 00:49:41]
$ sudo wireshark
[sudo] password for vboxuser:
** (wireshark:10237) 00:50:40.231501 [GUI WARNING] -- QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
** (wireshark:10237) 00:50:45.945446 [Capture MESSAGE] -- Capture Start ...
** (wireshark:10237) 00:50:46.002706 [Capture MESSAGE] -- Capture started
** (wireshark:10237) 00:50:46.002886 [Capture MESSAGE] -- File: "/tmp/wireshark_enp0s3I3Q4X2.pcapng"
^C
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 02:46:12]
$ sysctl net.ipv4.tcp_syncookies
net.ipv4.tcp_syncookies = 1
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 02:47:15]
$ sudo sysctl -w net.ipv4.tcp_syncookies=0
[sudo] password for vboxuser:
net.ipv4.tcp_syncookies = 0
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 02:53:38]
$ sysctl net.ipv4.tcp_syncookies
net.ipv4.tcp_syncookies = 0
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 02:53:44]
$ 
```

b)

DOS (Denial-of-Service) attack software is designed to flood a target system with an overwhelming volume of network requests, rendering it unable to respond to legitimate traffic. Among many tools available, hping3 stands out due to its flexibility and wide adoption in network testing. This hping can be used it in a flexible and customizable way. The main features are: It supports customizable packet crafting – means it allows precise control over packet headers and flags. Multi-protocol support – Works with TCP, UDP, ICMP, and raw IP Packets. High throughput – Can send packet at very high rate, making it suitable for SYN flood attacks.

The setup:

VM1: Target Machine running on server TCP

VM3: Attacker machine equipped with hping3

VM Network: Both Machines are on same sub network

Install DOS Attack Software: commands

In vm3:

sudo apt-get update

sudo apt-get install hping3

```

Nov 25 03:41
vboxuser@sruthi-vm3:~$ nano ./bashrc
vboxuser@sruthi-vm3:~$ nano ./bashrc
vboxuser@sruthi-vm3:~$ source ~/.bashrc
[sruthi-mandalapu vboxuser sruthi-vm3 ~ 2024-11-24 20:15:32]
$ sudo apt-get update
[SRU] http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:2 http://us.archive.ubuntu.com/ubuntu noble InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:4 http://security.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:5 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [7,192 kB]
Get:6 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [212 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [51,9 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [131 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [208 kB]
Get:10 http://us.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 kB]
Get:11 http://us.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [309 kB]
Get:12 http://us.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [208 kB]
Get:13 http://us.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 kB]
Get:14 http://us.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [212 kB]
Get:15 http://us.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [21,9 kB]
Get:16 http://us.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 kB]
Fetched 902 kB in 2s (461 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
[sruthi-mandalapu vboxuser sruthi-vm3 ~ 2024-11-25 03:39:44]
$ sudo apt-get install hping3
Reading package lists... Done
Building dependency tree... Done
Building state information... Done
The following NEW packages will be installed:
  hping3
0 upgraded, 1 newly installed, 0 to remove and 90 not upgraded.
Need to get 100.0 kB of archives.
After this operation, 237 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu noble/universe amd64 hping3 amd64 3.a2.ds2-10build2 [100.0 kB]
Fetched 100.0 kB in 0s (93.4 kB/s)
Selecting previously unselected package hping3.
(Reading database ... 148080 files and directories currently installed.)
Preparing to unpack .../hping3_3.a2.ds2-10build2_amd64.deb ...
Unpacking hping3 (3.a2.ds2-10build2) ...
Setting up hping3 (3.a2.ds2-10build2) ...
Processing triggers for man-db (2.12.0-4build2) ...
[sruthi-mandalapu vboxuser sruthi-vm3 ~ 2024-11-25 03:40:50]

```

Launch the SYN Flood Attack: commands

sudo hping3 -S -p 22 --flood 192.168.118.169 → where ip-address of vm1 is 192.168.118.169

```

Nov 25 04:03
vboxuser@sruthi-vm3:~$ sudo hping3 -S -p 22 --flood 192.168.118.169
hping3: missing host argument
Try 'hping3 -help' for more information.
[sruthi-mandalapu vboxuser sruthi-vm3 ~ 2024-11-25 03:46:32]
$ sudo hping3 -S -p 22 --flood 192.168.118.169
[sudo] password for vboxuser:
HPING 192.168.118.169 (enp0s3 192.168.118.169): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown

```

c)

Software Used:

The software I have used for the attack is hping software. Hping is a popular command-line tool often used for network testing. It can craft custom TCP, UDP, ICMP and raw IP packets. Hping is helpful in Firewall testing, Network Analysis. Misusing hping sends large volumes of packets can cause DOS attacks.

Working of Hping:

For a SYN Flood attack, it sends a huge number of TCP SYN packets to the target. Hence, leading to make resources by creating a half-open connections. TCP Handshake works as: SYN – Initiates a connection, SYN-ACK – Server responds to the request. ACK – Client acknowledges, completing the handshake. In SYN Flood

attack only SYN packets are sent, leaving the server waiting indefinitely for the final ACK, leading to resource exhaustion.

Steps for illustrating functionality of hping:

1. Setting up: Install hping – use sudo apt-get install hping3;

2. Crafting packets:

- Protocols: hping allows you to customize various aspects of network packets – TCP: -s for SYN, -A for ACK; UDP --udp.
- Source/Destination IPs and Ports: -a <spoofed_ip> : Spoof the source ip, -p <port> : Set destination port.
- Flags: SYN, ACK, FIN, RST, PSH (used for TCP handshakes and teardown)
- Payload: Attach data or customize packet size with -d
- Rate control: Limit packet sending speed using --flood or -i <interval>

3. Ping and Trace Routes: Simulate high traffic to analyse

Parameters used for hping:

sudo hping3 -S -p 22 --flood 192.168.118.169 – This is the command used for hping request

-S → Specifies sending TCP SYN packets

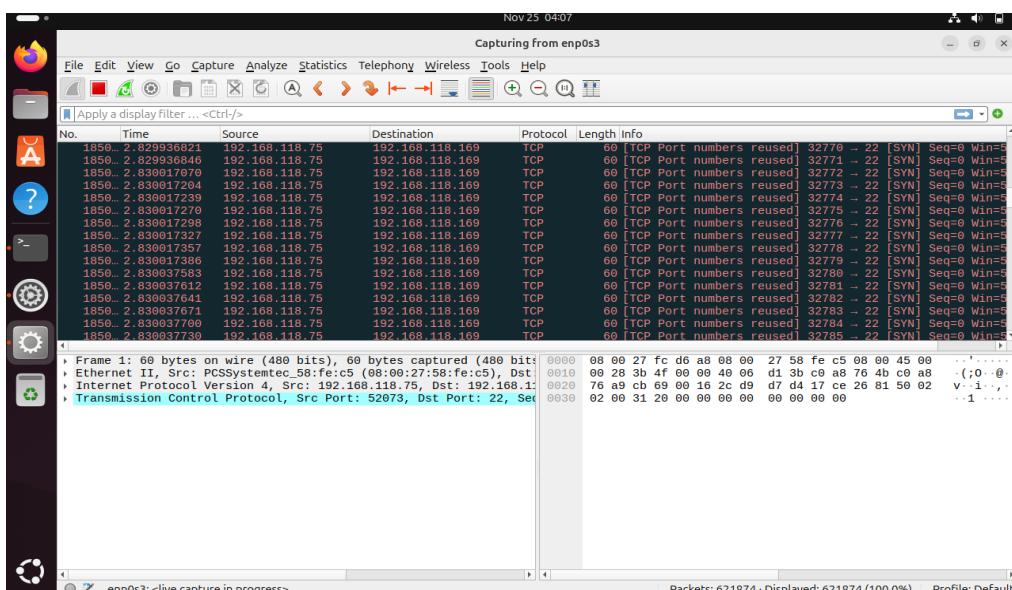
-p 22 → Target port 22 on the victim

--flood → Sends packets as fast as possible without waiting for replies

192.168.118.169 → IP Address of the target Machine (VM1)

d)

VM3 ip is 192.168.118.75, VM1 ip is 192.168.118.169 → we can observe continuous syn flood in VM1, and we are receiving from VM3. source ip - VM3, destination ip – VM1



e)

While attempting to establish an SSH session from VM2 to VM1, during ongoing SYN Flood attack, I have observed a delay and taking time waiting to connect, after sometime it couldn't able to connect finally shown up connection closed. Here, what happened is - SSH Session failed to connect with VM1. This is because the SYN Flood attack overwhelms VM1's networking stack, consuming resources like CPU and memory. It couldn't be completed as VM1 is too busy in processing or rejecting the flood of incoming SYN packets, leading unable to handle VM2 request.

ssh 192.168.118.169 → the ip-address of VM1

```
Nov 25 04:41  
Terminal  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
[sruthi-mandalapu vboxuser sruthi-vm2 ~ 2024-11-25 04:37:50]  
$ ssh 192.168.118.169  
Connection closed by 192.168.118.169 port 22  
[sruthi-mandalapu vboxuser sruthi-vm2 ~ 2024-11-25 04:40:51]  
$ █
```

f)

In first scenario when cookies are disabled, VM1 allocated resources for every incoming SYN packet. The backlog quickly fills up under SYN flood causing system unable to work. We can see in wireshark, it shows a large number of connections stuck in SYN_RECV state. In other scenario, when cookies are enabled, VM1 doesn't allocate resources for incoming SYN packets until handshake is completed. We can see in wireshark, it shows a few connections remain in SYN_RECV state.

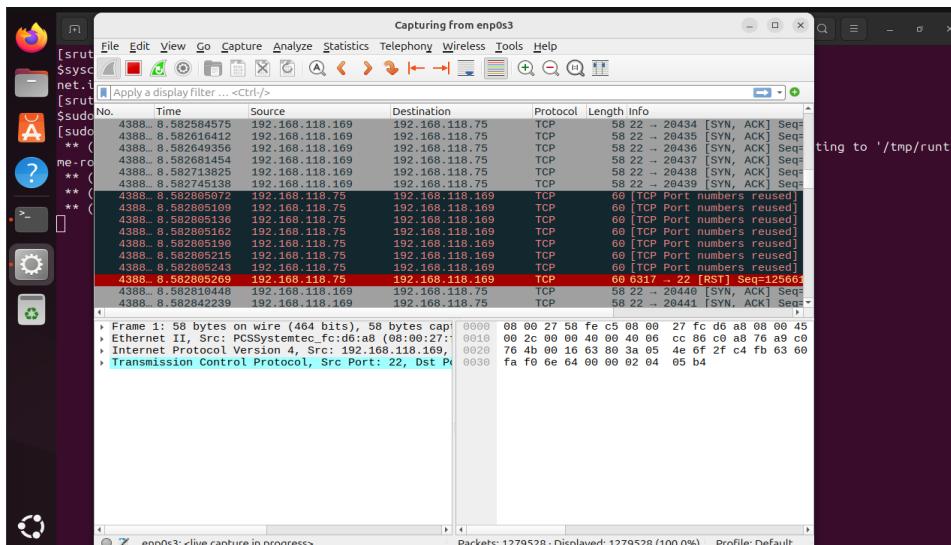
With cookies disabled: High volume of SYN packets, delayed or missing SYN-ACK responses, SYN-ACK packets often go unanswered.

With cookies enabled: High volume of SYN packets but still visible, SYN-ACK responses are sent promptly for all packets, ACK responses are processed allowing connections to complete.

When you see the screenshots in 1st scenario – most of the packets are SYN packets. In 2nd scenario – every SYN packet is completing a handshake protocol by producing acknowledgement responses as SYN-ACK.



```
Nov 25 05:23:37 [sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 05:21:05] $sysctl net.ipv4.tcp_syncookies net.ipv4.tcp_syncookies = 1 [sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 05:23:37] $
```

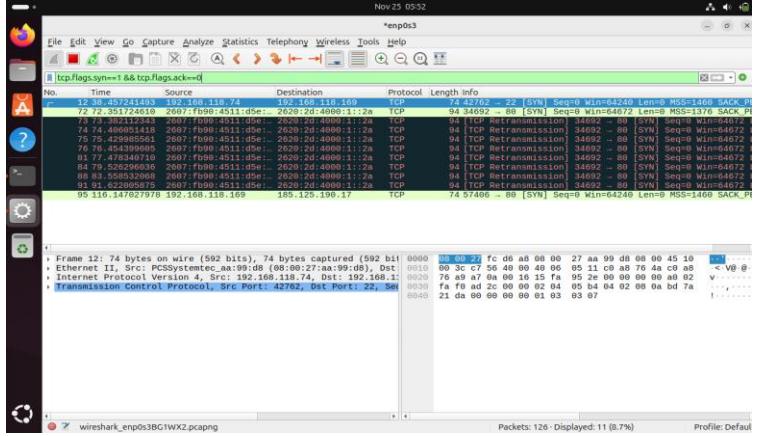


PART4: Traffic Analysis

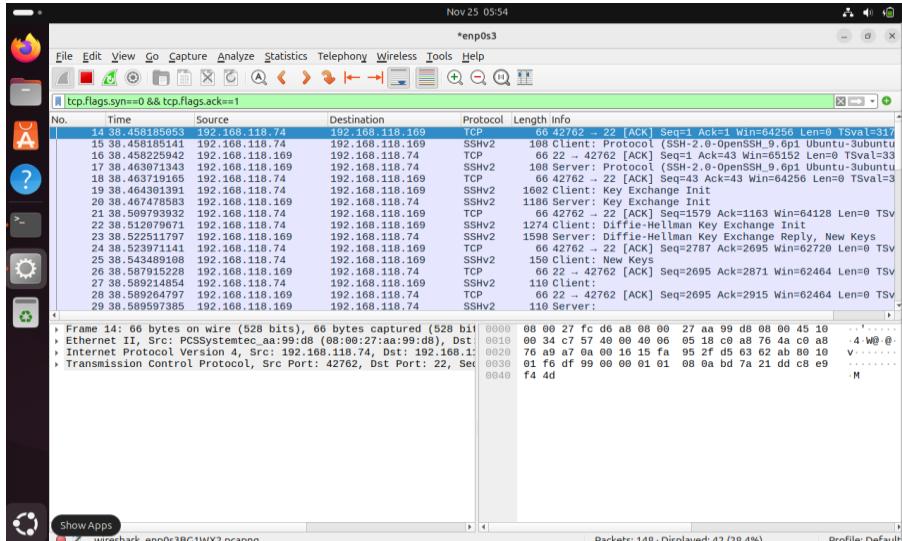
a)

Before SYN Flood:

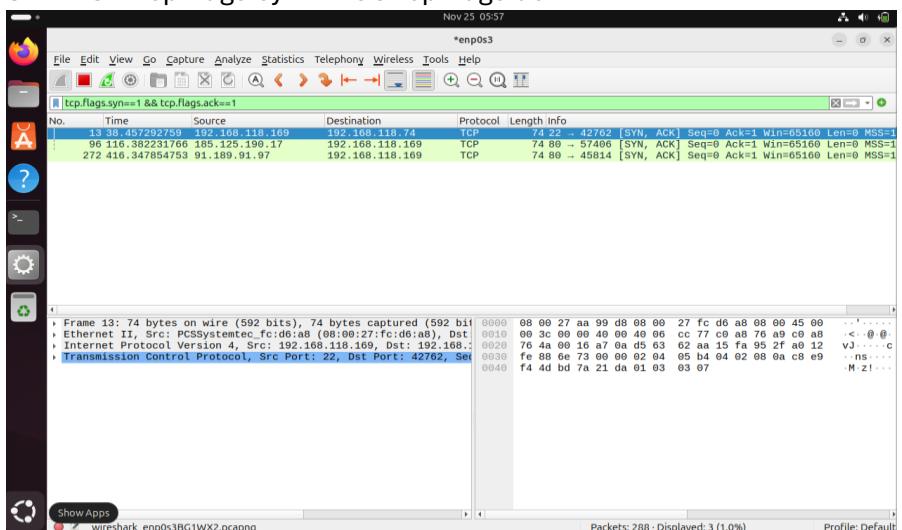
SYN: $\text{tcp.flags.syn} == 1 \& \& \text{tcp.flags.ack} == 0$



ACK: $\text{tcp.flags.syn} == 0 \& \& \text{tcp.flags.ack} == 1$



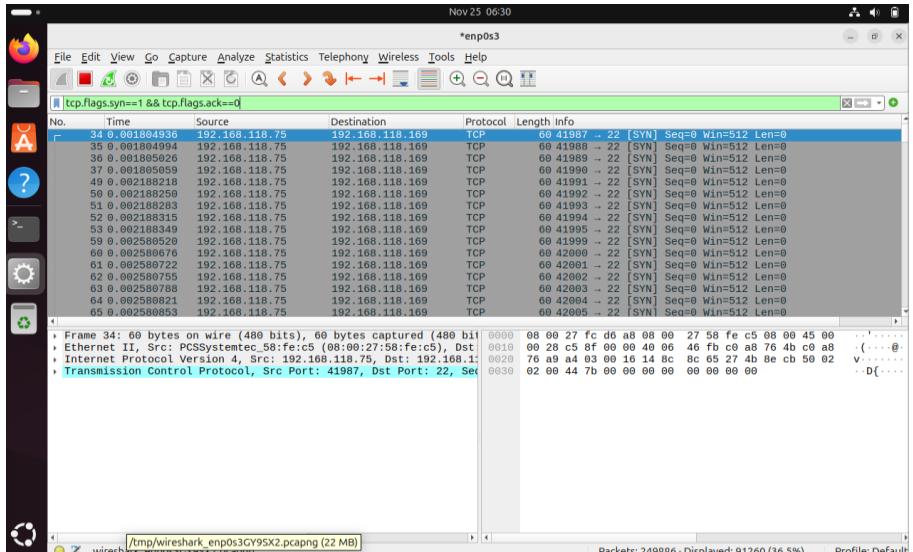
SYN-ACK: $\text{tcp.flags.syn} == 1 \& \& \text{tcp.flags.ack} == 1$



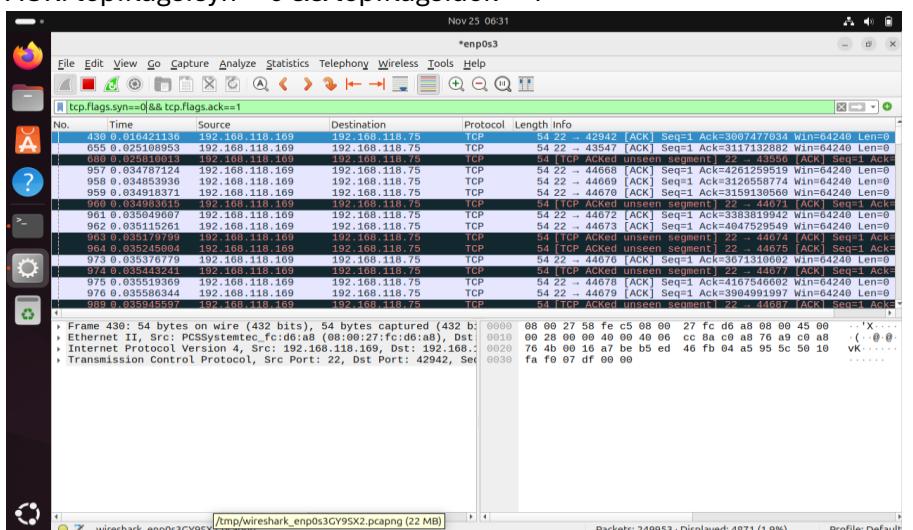
SYN Flood when SYN Cookies enabled:

By default, the cookies are enabled

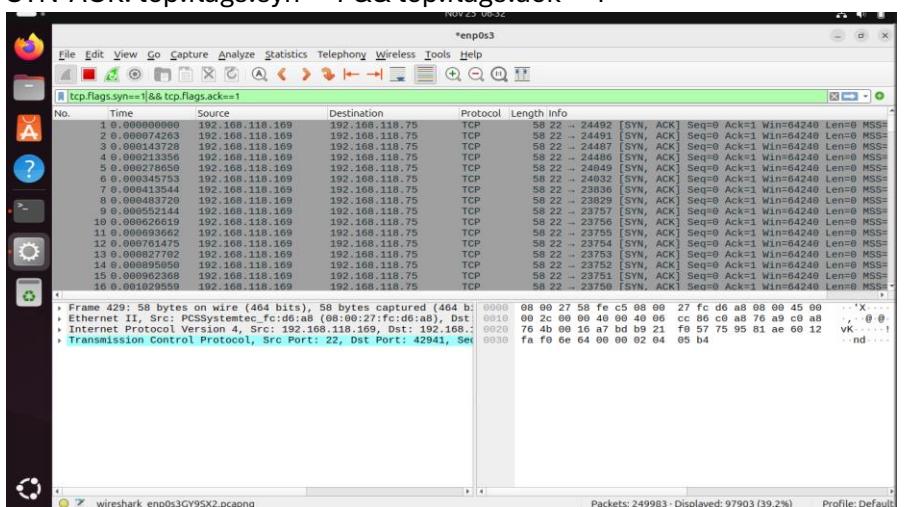
SYN: $\text{tcp.flags.syn} == 1 \& \& \text{tcp.flags.ack} == 0$



ACK: $\text{tcp.flags.syn} == 0 \& \& \text{tcp.flags.ack} == 1$



SYN-ACK: $\text{tcp.flags.syn} == 1 \& \& \text{tcp.flags.ack} == 1$



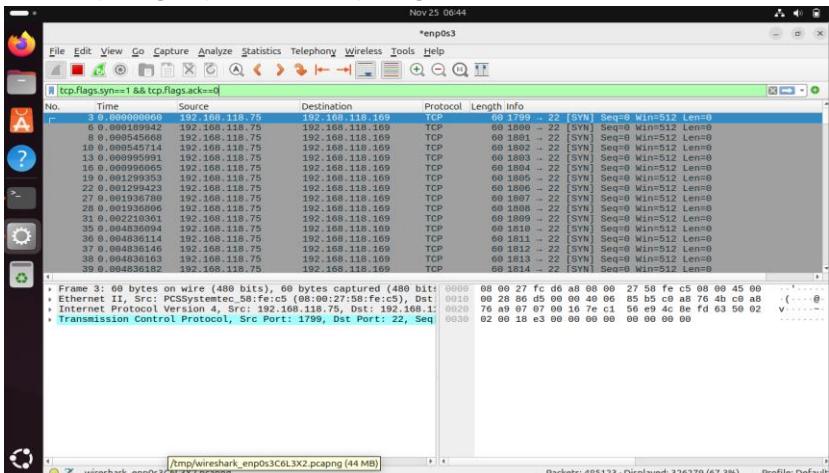
SYN Flood when SYN Cookies disabled:

```

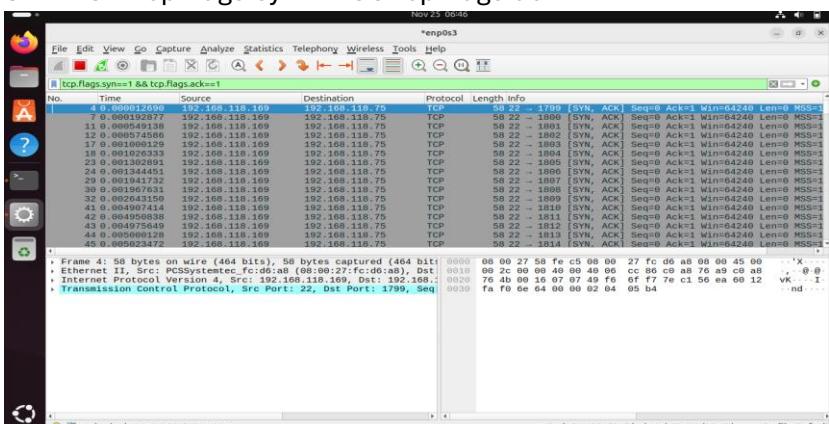
Nov 25 06:42
Terminal
** (wireshark:3501) 06:20:01.000630 [Capture MESSAGE] -- Capture started
** (wireshark:3501) 06:20:01.000697 [Capture MESSAGE] -- File: "/tmp/wireshark_enp0s3870RX2.pcapng"
Killed
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 06:27:21]
$ sudo wireshark
** (wireshark:3599) 06:28:16.223388 [GUI WARNING] -- QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
** (wireshark:3599) 06:28:26.791110 [Capture MESSAGE] -- Capture Start ...
** (wireshark:3599) 06:28:27.069261 [Capture MESSAGE] -- Capture started
** (wireshark:3599) 06:28:27.069672 [Capture MESSAGE] -- File: "/tmp/wireshark_enp0s3GY95X2.pcapng"
** (wireshark:3599) 06:35:10.665114 [Capture MESSAGE] -- Capture Stop ...
** (wireshark:3599) 06:35:10.696862 [Capture MESSAGE] -- Capture stopped.
** (wireshark:3599) 06:35:10.697115 [Capture WARNING] ./ui/capture.c:72 -- capture_input_closed():
^C
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 06:35:19]
$ sysctl net.ipv4.tcp_synccookies
net.ipv4.tcp_synccookies = 1
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 06:39:31]
$ sysctl -w net.ipv4.tcp_synccookies=0
sysctl: permission denied on key "net.ipv4.tcp_synccookies", ignoring
net.ipv4.tcp_synccookies = 0
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 06:40:04]
$ sudo sysctl -w net.ipv4.tcp_synccookies=0
[sruthi-mandalapu vboxuser sruthi-vm1 ~ 2024-11-25 06:40:45]
$ sudo wireshark
** (wireshark:3692) 06:41:28.111289 [GUI WARNING] -- QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
** (wireshark:3692) 06:41:35.493839 [Capture MESSAGE] -- Capture Start ...
** (wireshark:3692) 06:41:35.549409 [Capture MESSAGE] -- Capture started
** (wireshark:3692) 06:41:35.549551 [Capture MESSAGE] -- File: "/tmp/wireshark_enp0s3C6L3X2.pcapng"

```

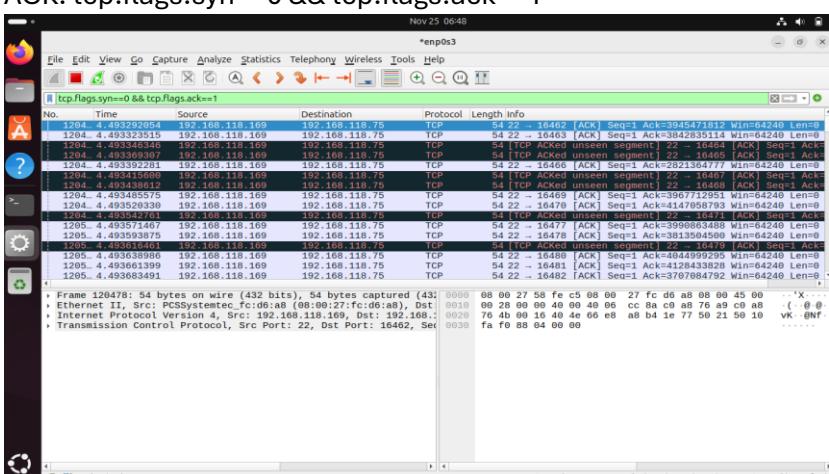
SYN: tcp.flags.syn==1 && tcp.flags.ack==0



SYN-ACK: tcp.flags.syn==1 && tcp.flags.ack==1



ACK: tcp.flags.syn==0 && tcp.flags.ack==1



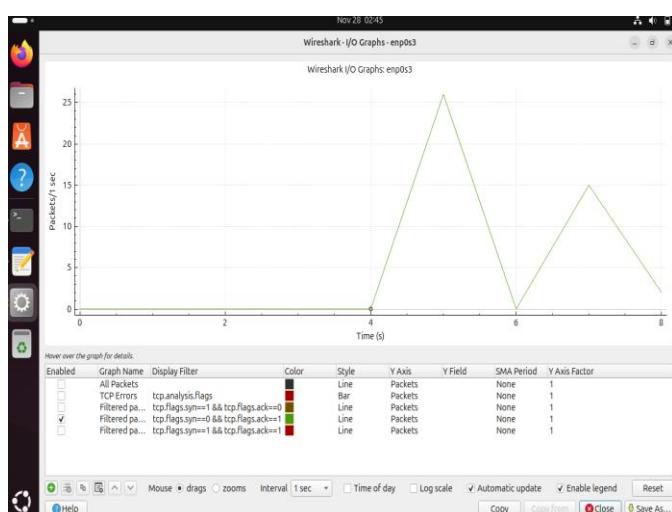
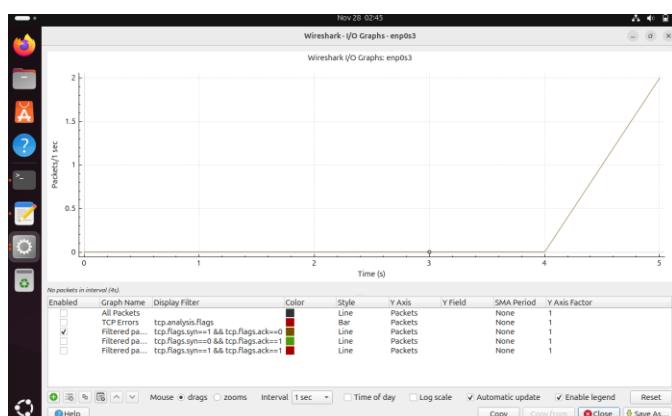
As you can see in the below table – Displayed percentage in wireshark at bottom – among all the three scenarios:

| | Before Flood Attack | Cookies Enabled - Flood | Cookies Disabled - Flood |
|---------|---------------------|-------------------------|--------------------------|
| SYN | 8.7% | 36.25% | 67.3% |
| SYN-ACK | 1% | 39.2% | 16.4% |
| ACK | 28.4% | 1.9% | 0.1% |

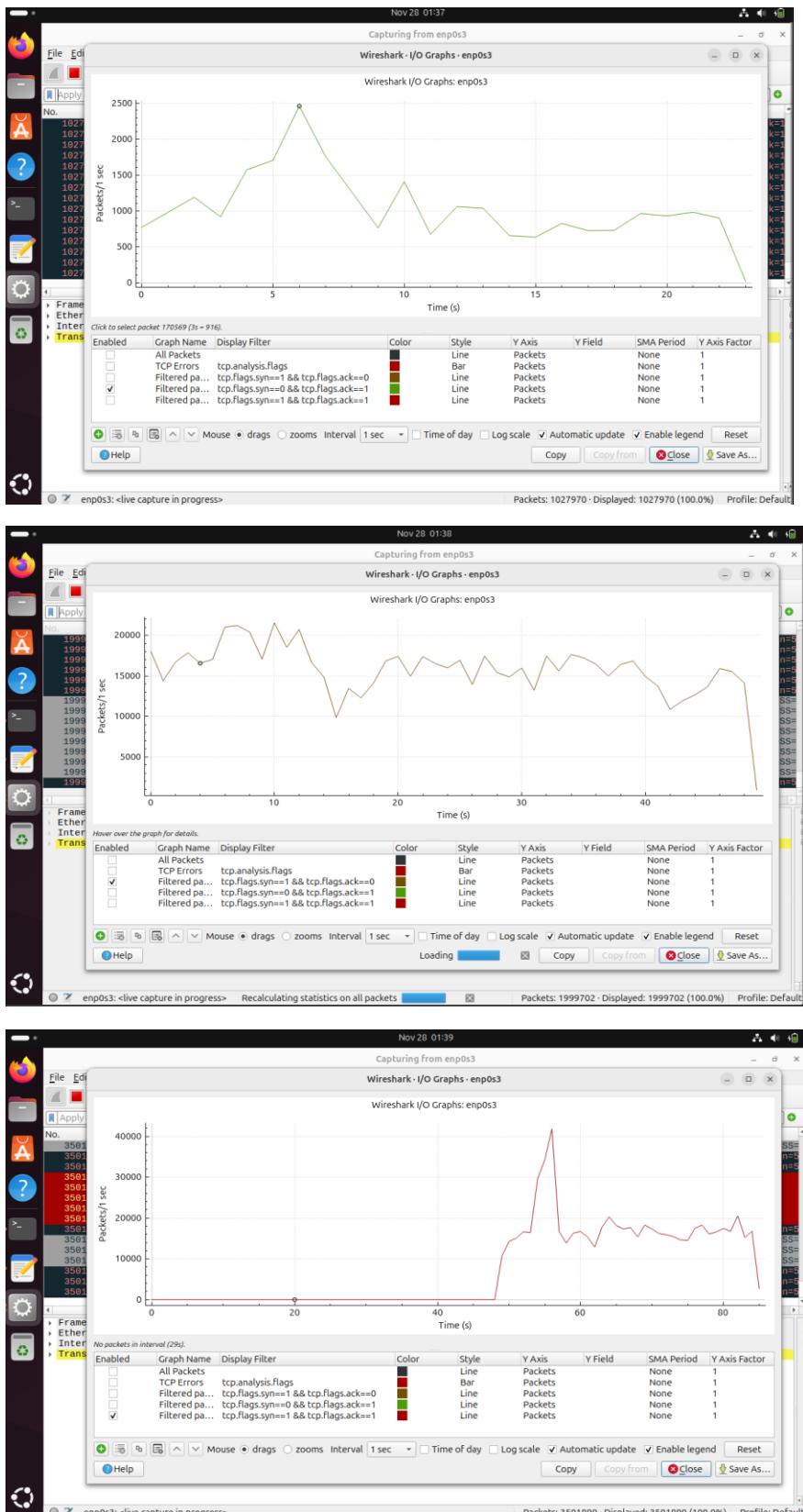
As you can see before flood attack, almost equal proportions for syn, syn-ack, ack. As it is not a flood attack you could only observe 8 percent as SYN packets. In case of Cookies enabled and flood attack. As this is flood attack – the SYN is increased from 8 percent to 36.25 percent. Until the acknowledgement is received, the vm waits to finish up each SYN packet. Hence, SYN and SYN-ACK almost have equal proportions (each packet waits to finish handshake protocol). When cookies is disabled, the SYN packet keeps on generating – hence there is increase in SYN packets received than before – it is from 36% to 67%. And it couldn't able to give the responses hence there is low SYN-ACK (which is 16% reduced than before).

b)

Before Flood Attack: (Made ssh connection from vm2 to vm1)

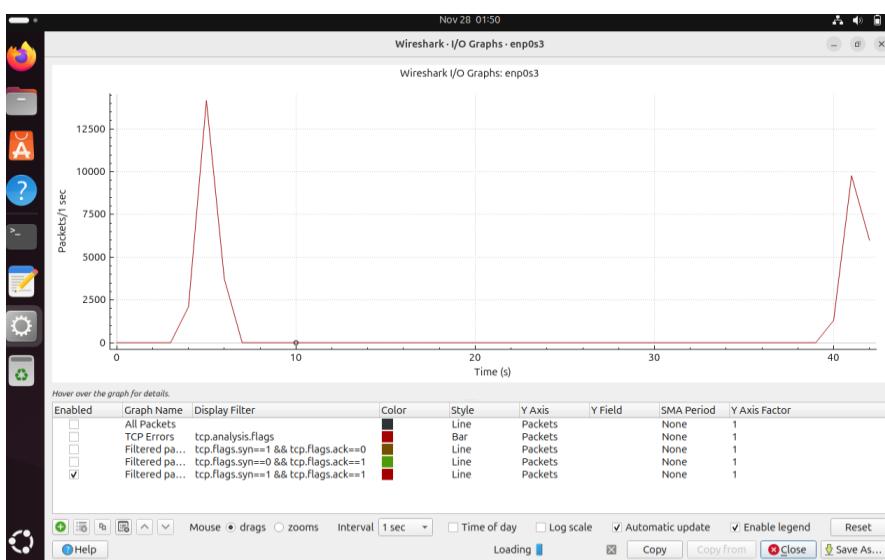
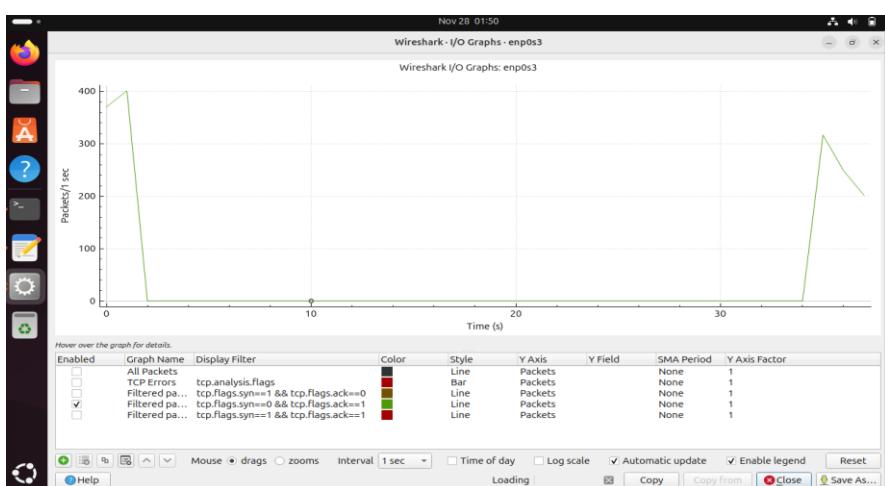
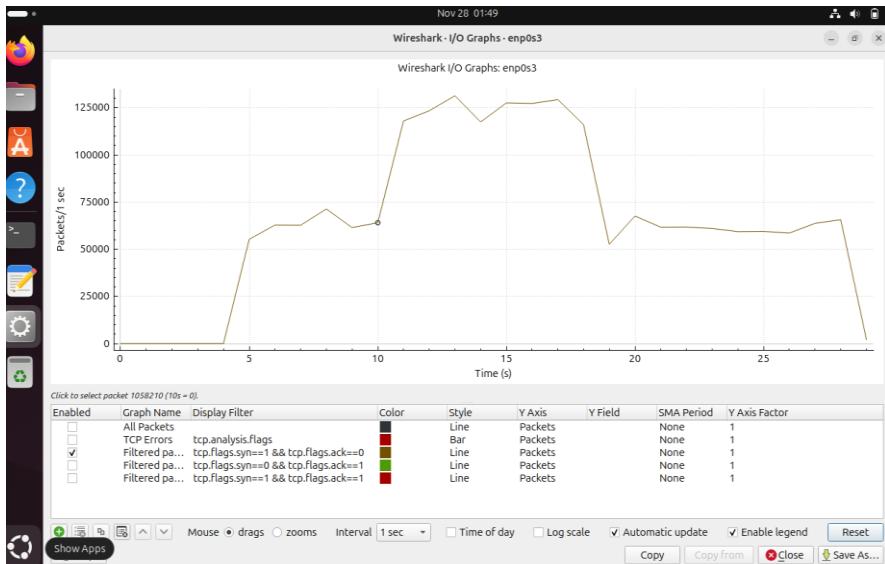


During Flood Attack when cookies is enabled:



When the cookies are enabled – we will finish handshake protocol for every packet, rather than allocating all the packets that was sent from the attacker. So, we can observe the SYN, SYN-ACK and ACK packets capturing over times. Firstly, SYN is sent then SYN-ACK is sent. As it is a flood attack – attacker focuses on sending the packets, hence there are no ACK first few seconds, after sometime we could find ACK packets.

During flood attack when cookies is disabled:



When cookies are enabled, the main focus is on transmitting the packets, it allocates all the packets firstly rather than completing the connections. Which explains leaving too many open connections. Hence, we find high data for SYN packets. Coming to SYN-ACK, as it focuses completely on allocating all the receiving packets, hence we have less data rate for SYN-ACK. And almost very much less packets that is almost negligible packets for ACK.

Data Rate:

Data Rate = Total number of packets transmitted/Time (in seconds)

- Data Rate for before attack: (for first 10 seconds)
Data Rate = $(0+0+0+0+0+2+0+0+0+0)/10 = 0.2$ packets/sec
- During Attack, when cookies is enabled: (for first 10 seconds)
Data Rate = $(750+1000+1200+1500+1000+1700+2500+2200+1700+750+1400)/10 = 1500$ packets/sec
(approximation – as a rough values from graph)
- During Attack, when cookies is disabled: (for first 5 seconds)
Data Rate = $(0+0+0+0+60000+65000+65000+70000+60000+60000+65000)/10 = 44000$ packets/sec
(approximation – as a rough values from the graph)

Data Rates for incoming packets to VM1 – this explains about the SYN packets. As you can see from above data rates, before attack the data rate is too low, when the attack but when cookies is enabled the data rate is little higher than before attack scenario. Lastly, when we check when cookies are disabled it was too high, projecting maximal data rates comparison to the above 2 conditions.

c)

As you can see in the below table – among all the three scenarios:

Table taken from Part (a) – figures (at the bottom – there is a packet count and filter count and the percentage)

| | Before Flood Attack | Cookies Enabled - Flood | Cookies Disabled - Flood |
|---------------------|---------------------|-------------------------|--------------------------|
| Total no of packets | 200 (Round figure) | 2,50,000 (Round figure) | 5,00,000 (Round Figure) |
| SYN Packets | 11 (8.7%) | 92,000 (36.25%) | 3,30,000 (67.3%) |
| SYN-ACK Packets | 3 (1%) | 97,000 (39.2%) | 80,000 (16.4%) |

The above table explains there is a baseline data (before flood attack), during flood attack – cookies enabled, during flood attack – cookies disabled. The table data explains about count of packets and the percentage. SYN packets explain about SYN_SENT, SYN-ACK packets explain about SYN_RECEIVED. Data is displayed based on before attack, attack on cookies enabled, attack on cookies disabled at specified instance that is (SYN, SYN-ACK) is displayed. We can find the number packets at each state (that is SYN/SYN-ACK) and also the percentage. We can observe clearly before attack, followed by cookies enabled, followed by cookies disabled the total number of packets is strictly increasing. When SYN packets – even in this case the number of packets are strictly increasing. When SYN-ACK packets when its disabled (this state gives slight lower when compared to the state of enabled). Overall, the attack when cookies are disabled affects the most receiving the highest number of packets.