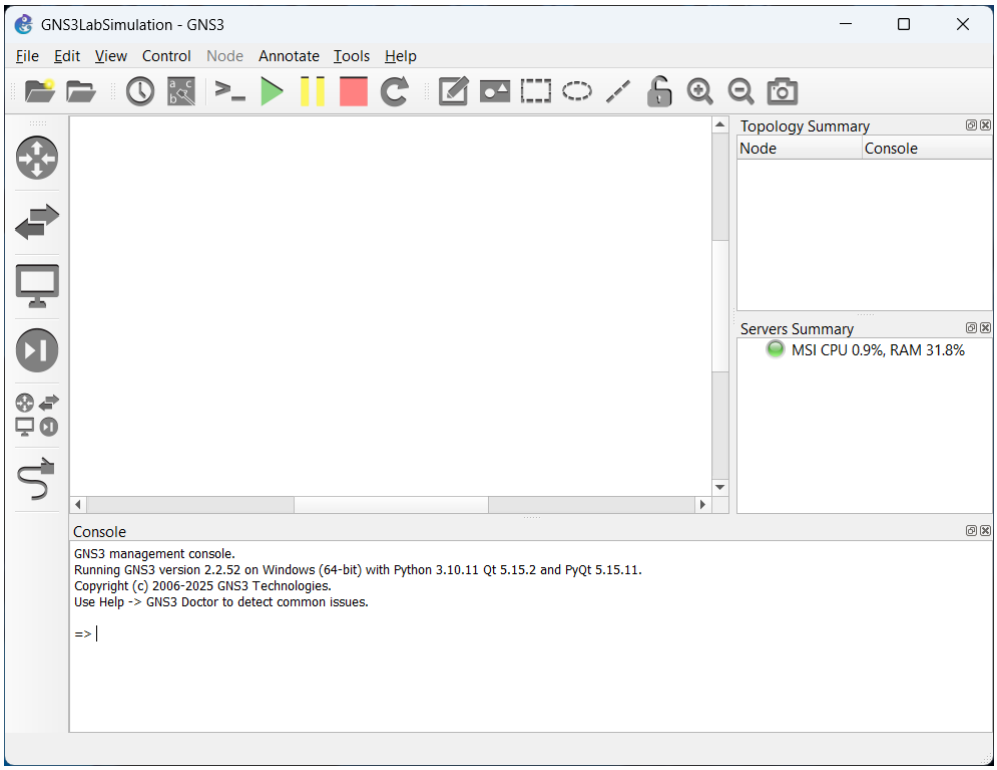
How to analyze the traffic captured in the Wireshark

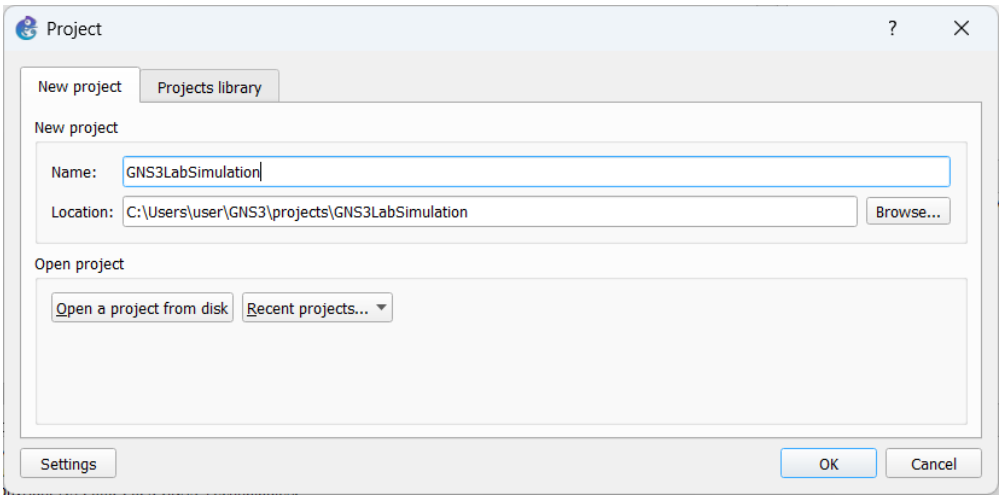
Environment setup (VMware and GNS3 configuration)

GNS3 setup

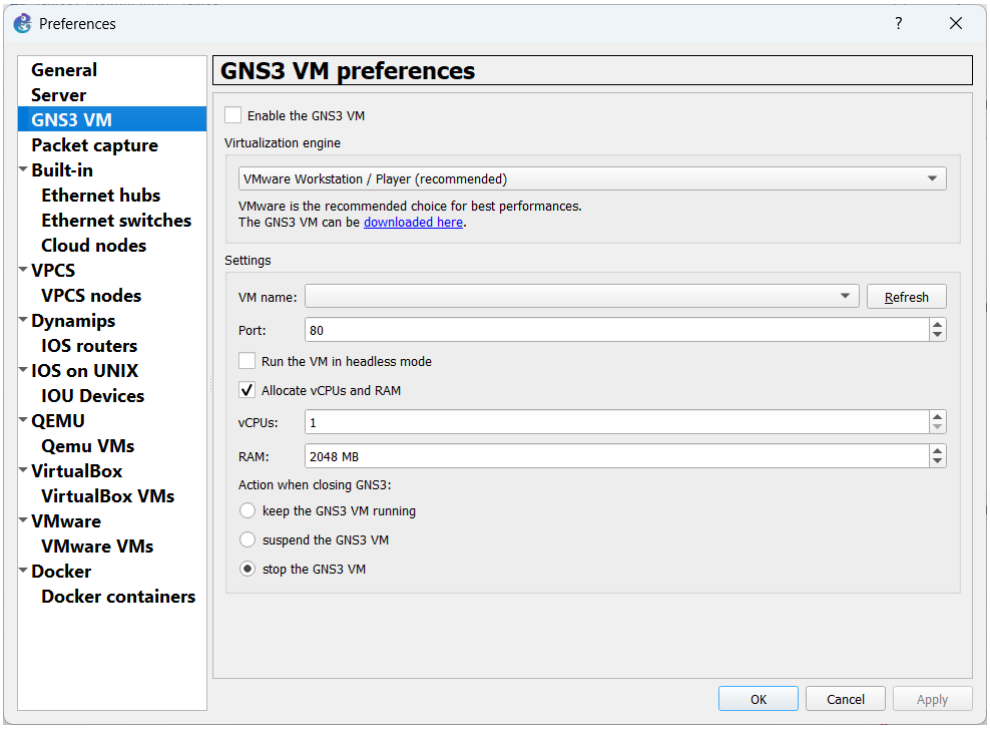
1. Download GNS3 from https://www.gns3.com/software/download
2. Open GNS3 from 'Downloads'
    a. You'll see this windows



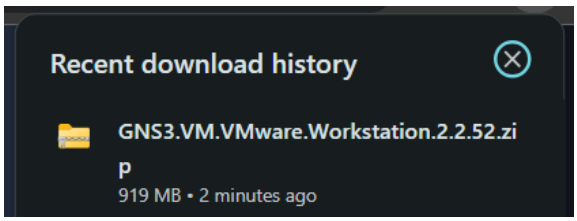
3. Name your project.
    a. Here it's going to be named 'GNS3LabSimulation'



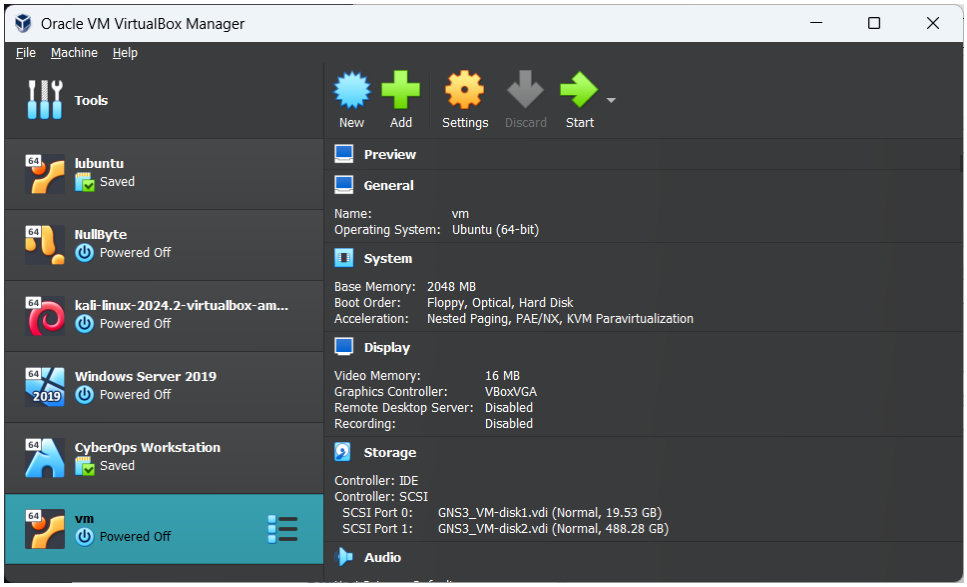
4. Open Edit >> Preferences >> GNS3 VM preferences
    a. You'll see this windows



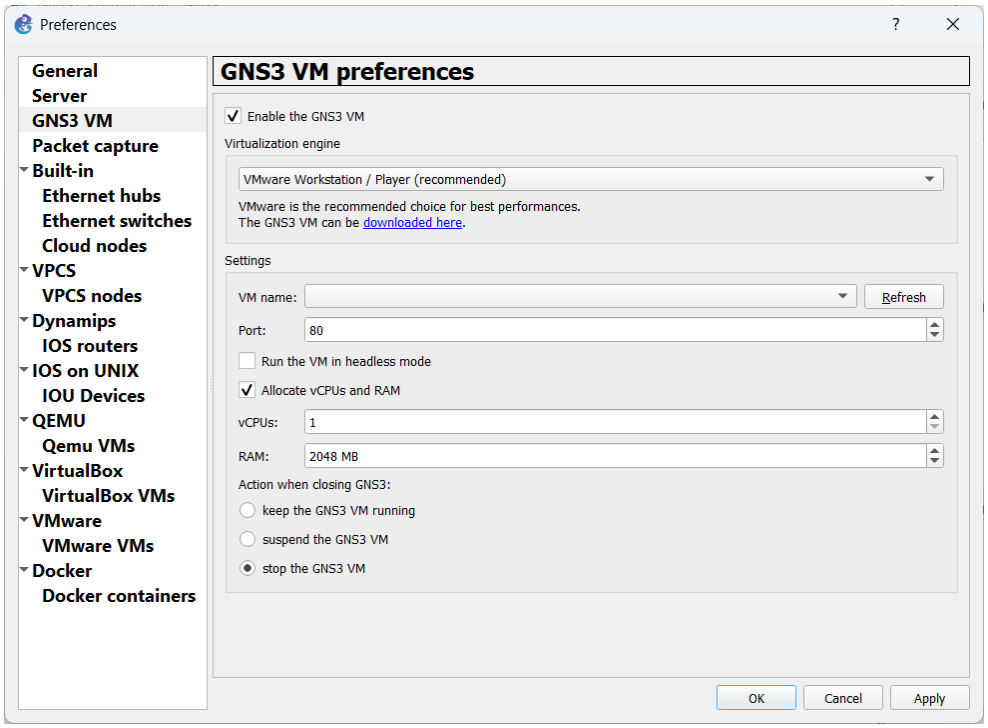
    b. Tap 'download here' >> download the file.
    c. Wait till the download finish.

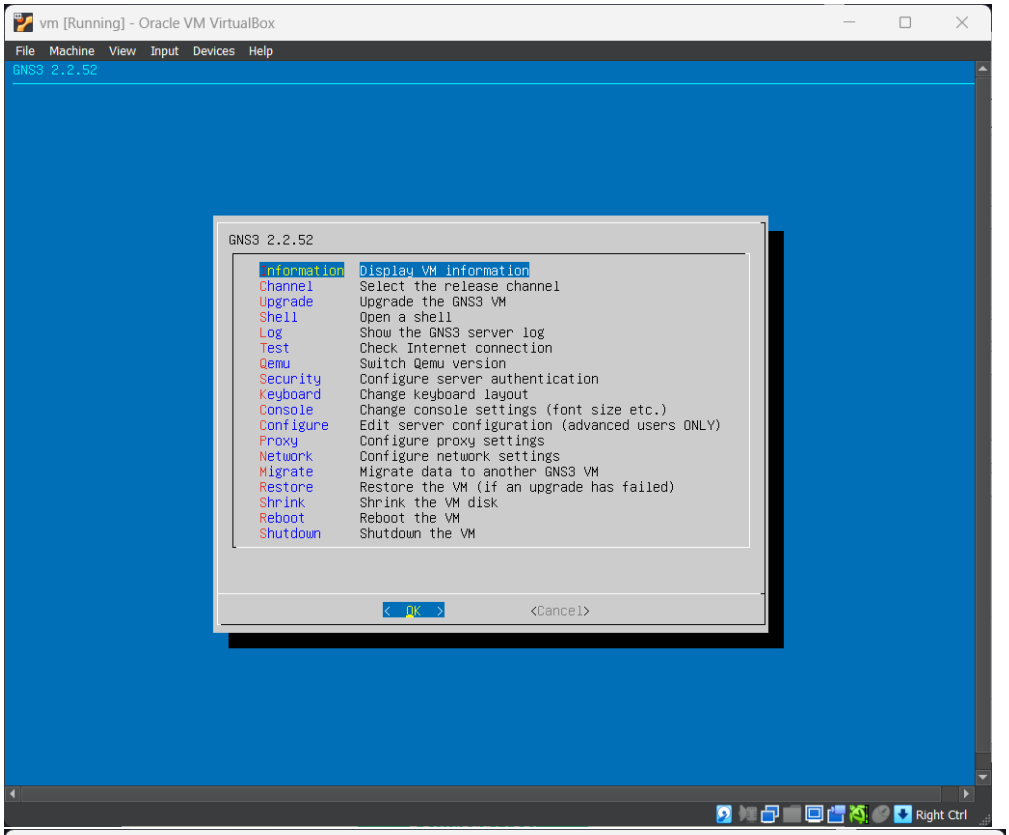


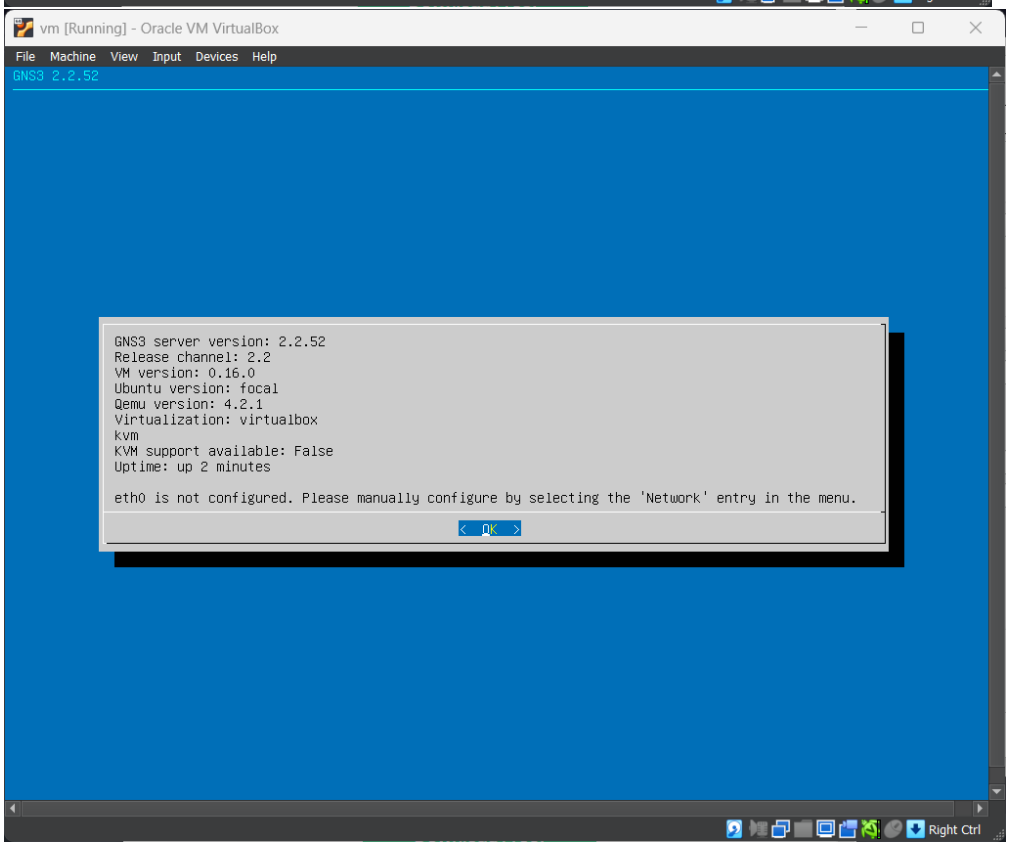
5. Open the file and it will be opened in your Oracle VM.



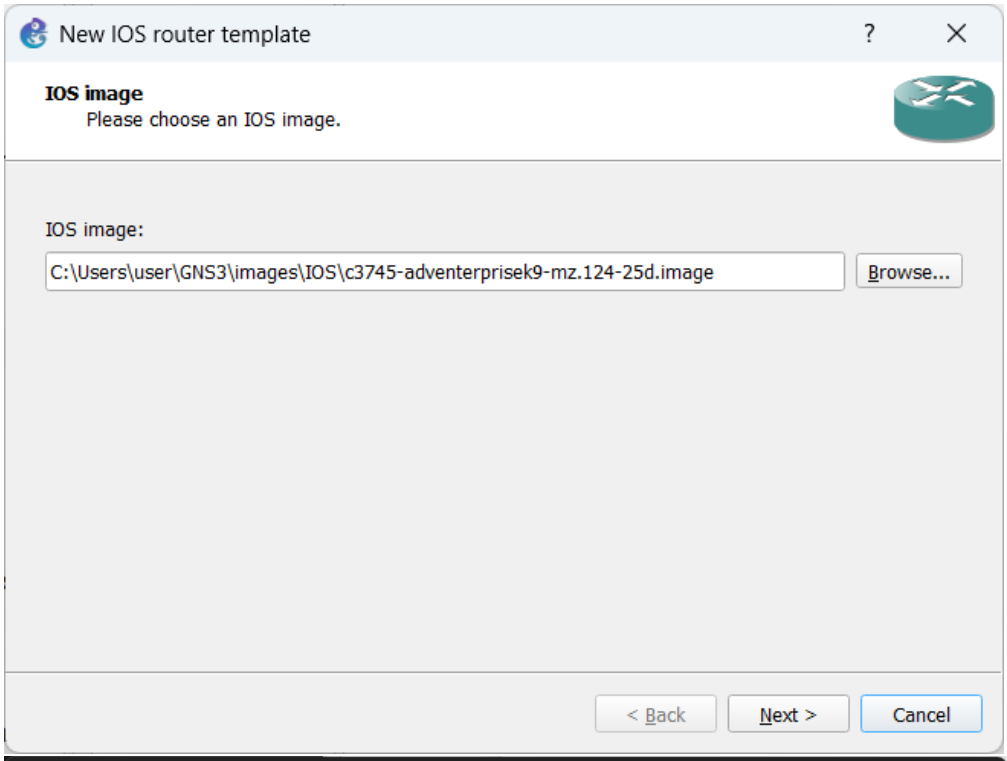
6. Go to your ‘GNS3 VM preferences’ >> Tap ‘Enable the GNS3 VM’ >> ‘Apply’
    a. You’ll be met with below windows :

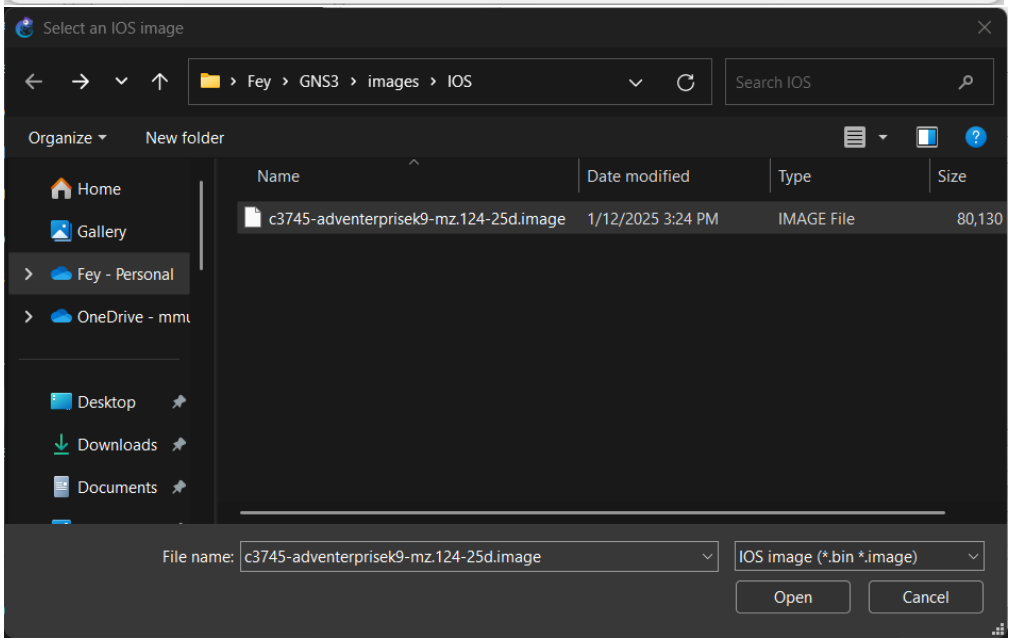



vm [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
GNS3 2.2.52
GNS3 2.2.52
Information Display VM information
Channel Select the release channel
Upgrade Upgrade the GNS3 VM
Shell Open a shell
Log Show the GNS3 server log
Test Check Internet connection
Qemu Switch Qemu version
Security Configure server authentication
Keyboard Change keyboard layout
Console Change console settings (font size etc.)
Configure Edit server configuration (advanced users ONLY)
Proxy Configure proxy settings
Network Configure network settings
Migrate Migrate data to another GNS3 VM
Restore Restore the VM (if an upgrade has failed)
Shrink Shrink the VM disk
Reboot Reboot the VM
Shutdown Shutdown the VM
< OK >
<Cancel>
Right Ctrl

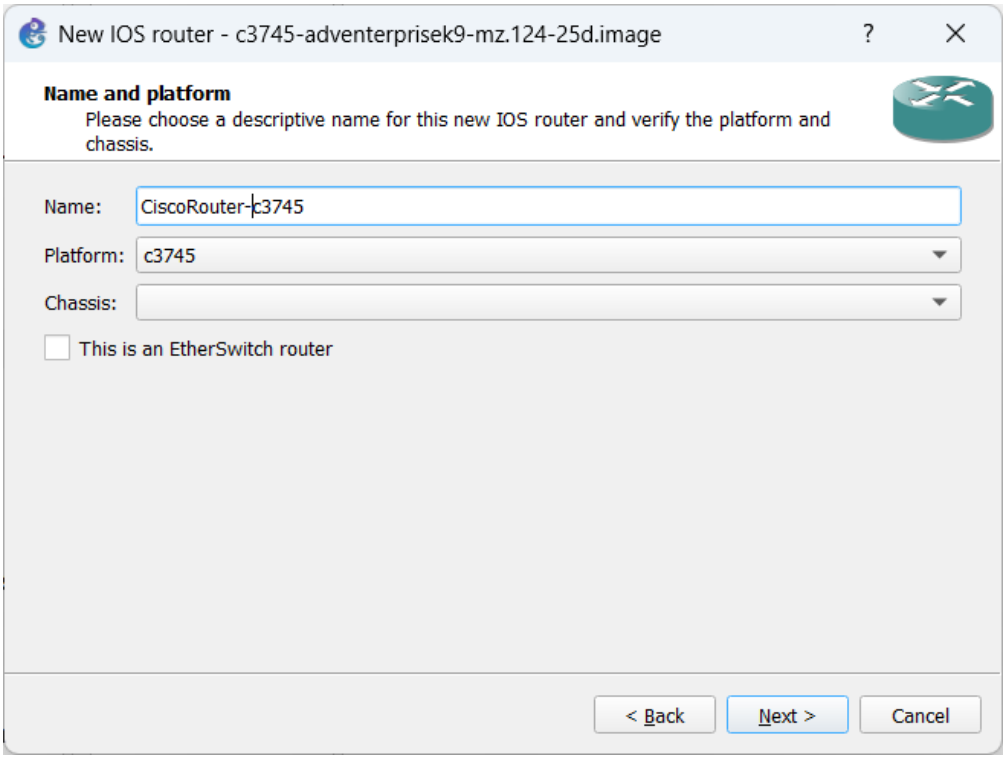

vm [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
GNS3 2.2.52
GNS3 server version: 2.2.52
Release channel: 2.2
VM version: 0.16.0
Ubuntu version: focal
Qemu version: 4.2.1
Virtualization: virtualbox
kvm
KVM support available: False
Uptime: up 2 minutes
eth0 is not configured. Please manually configure by selecting the 'Network' entry in the menu.
< OK >
Right Ctrl

7. In ‘GNS3 VM preferences’, go to ‘IOS routers’ >> ‘New’ >> ‘Browse’ >> choose your router IOS image that you have
    a. You can download from here too https://github.com/hegdepavankumar/Cisco-Images-for-GNS3-and-EVE-NG?tab=readme-ov-file
    b. Choose your IOS image

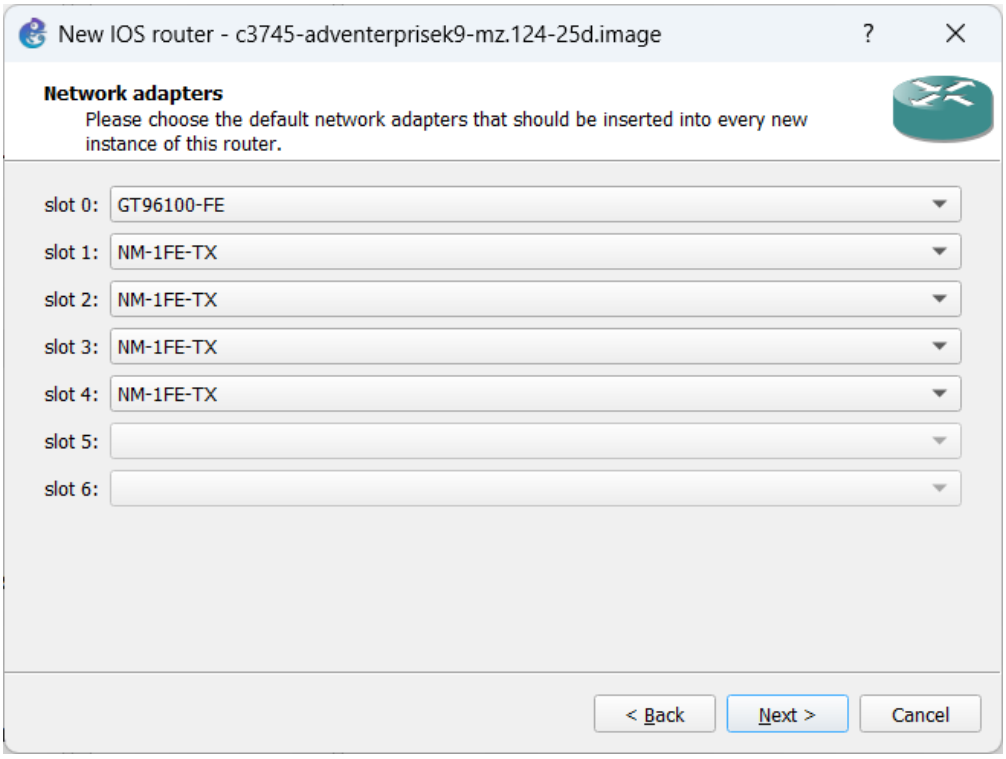




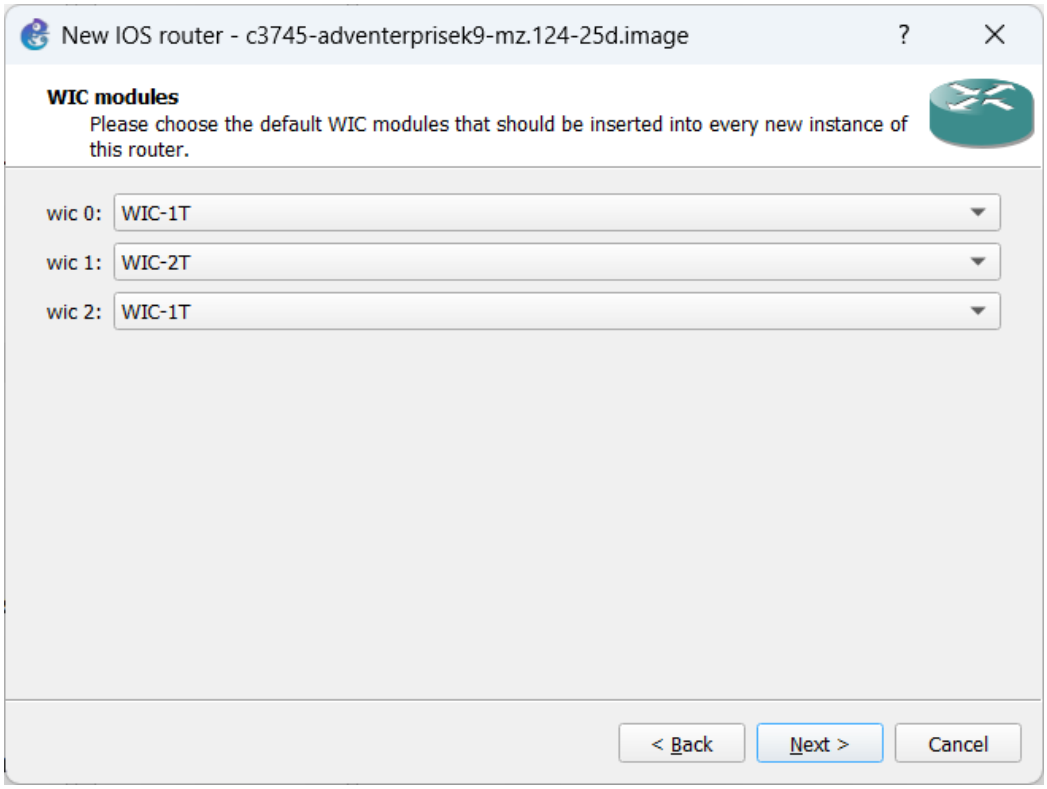
    c. Rename your router >> Next



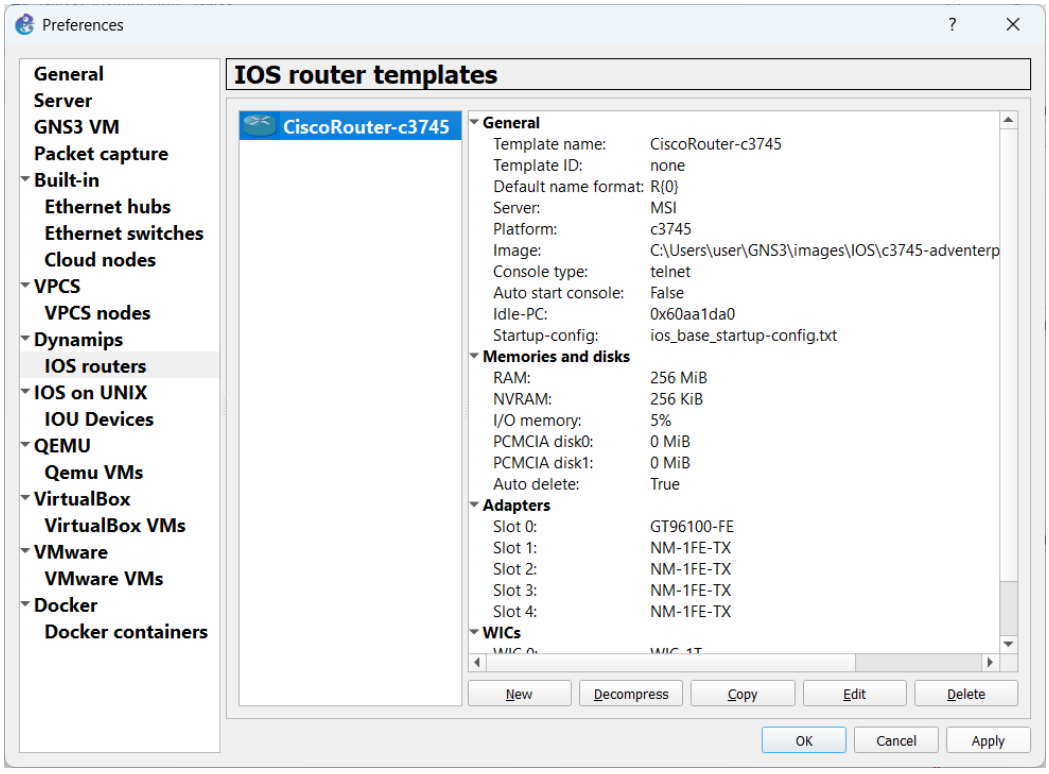
d. Specify your ports



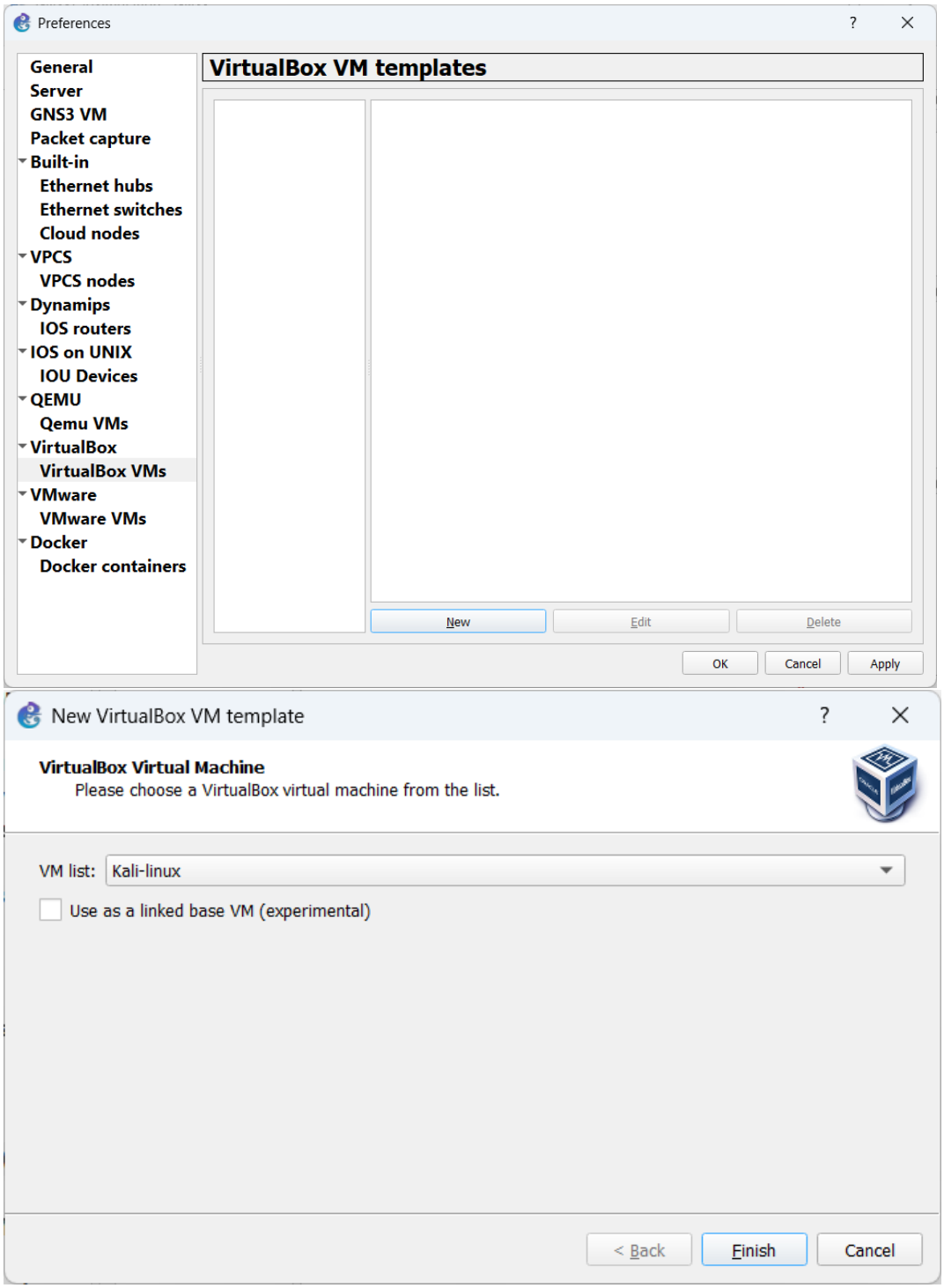
e. Specify your wireless ports



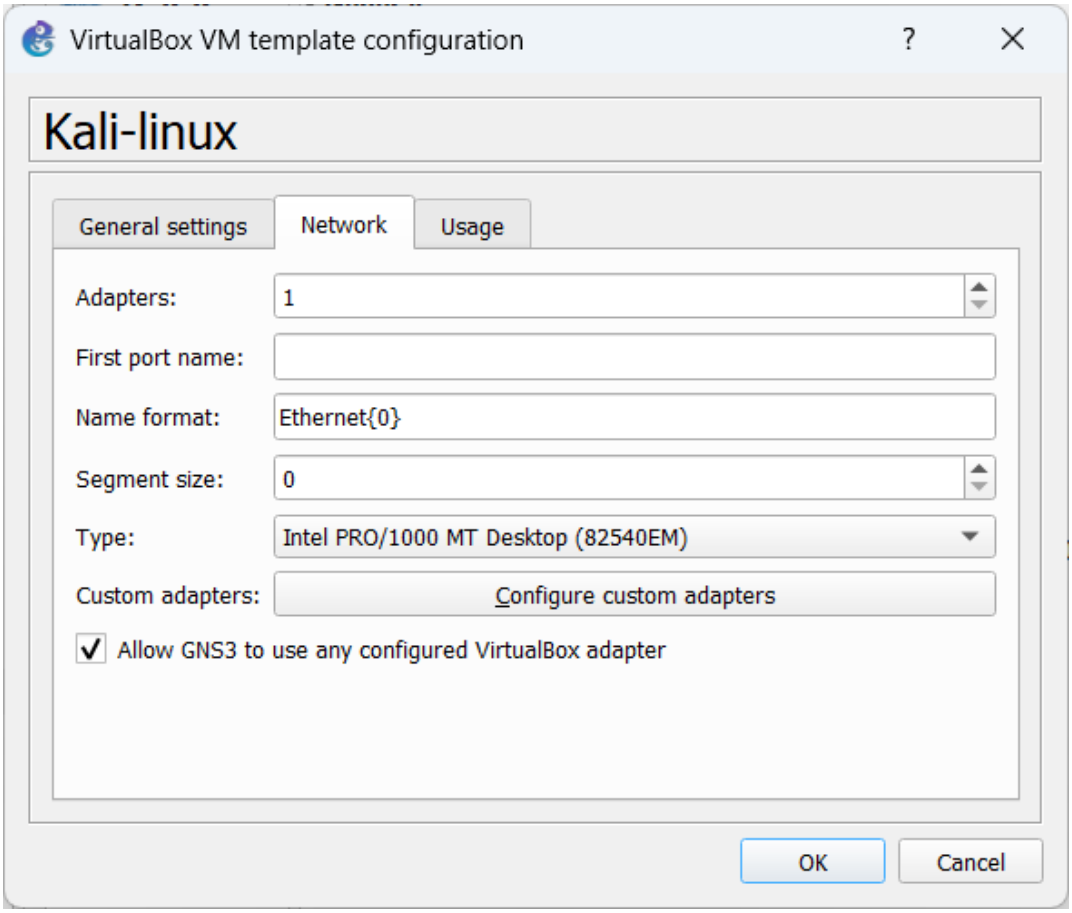
f. Next >> Finish >> Apply



8. In ‘GNS3 VM preferences’, go to ‘VirtualBox VMs’ >> New

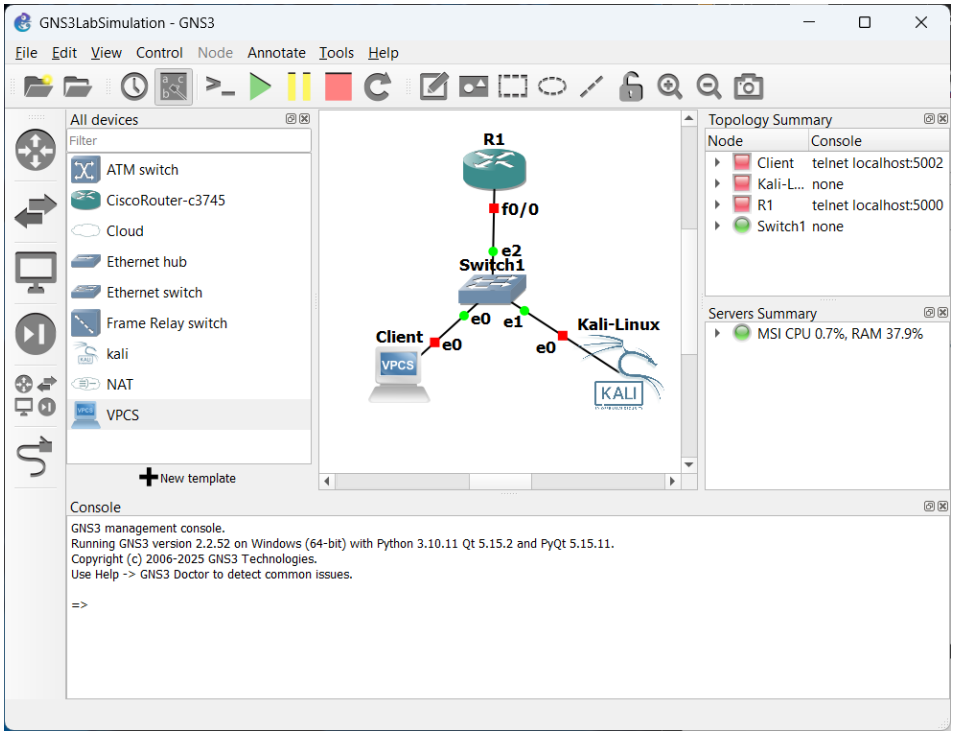


a. Tap on ‘Allow GNS3 to use any configured VirtualBox adapter’ >> OK >> Apply >> OK

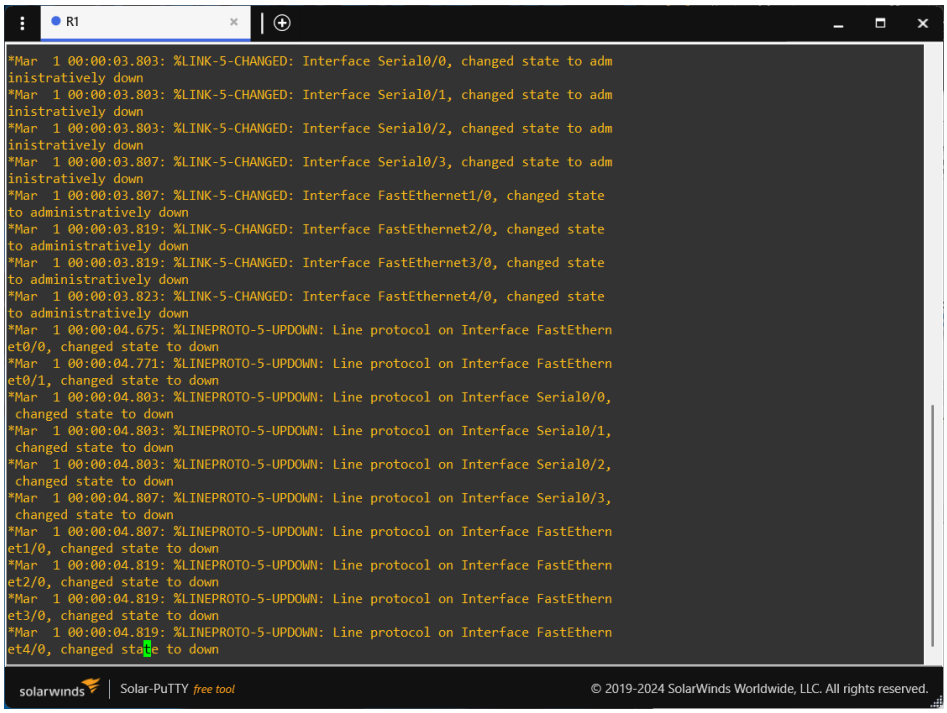

VirtualBox VM template configuration
Kali-linux
General settings
Network
Usage
Adapters:
1
First port name:
Name format:
Ethernet{0}
Segment size:
0
Type:
Intel PRO/1000 MT Desktop (82540EM)
Custom adapters:
Configure custom adapters
Allow GNS3 to use any configured VirtualBox adapter
OK
Cancel

Network topology creation in GNS3

1. Setup a simple topology between cisco router, switch, vpcs and kali



2. On R1, choose 'Start' >> Console
   a. You'll see this windows



   b. Show ip address

```
R1#sh ip int brief
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            unassigned      YES unset  administratively down down
Serial0/0                  unassigned      YES unset  administratively down down
FastEthernet0/1            unassigned      YES unset  administratively down down
Serial0/1                  unassigned      YES unset  administratively down down
Serial0/2                  unassigned      YES unset  administratively down down
Serial0/3                  unassigned      YES unset  administratively down down
FastEthernet1/0            unassigned      YES unset  administratively down down
FastEthernet2/0            unassigned      YES unset  administratively down down
FastEthernet3/0            unassigned      YES unset  administratively down down
FastEthernet4/0            unassigned      YES unset  administratively down down
R1#
```

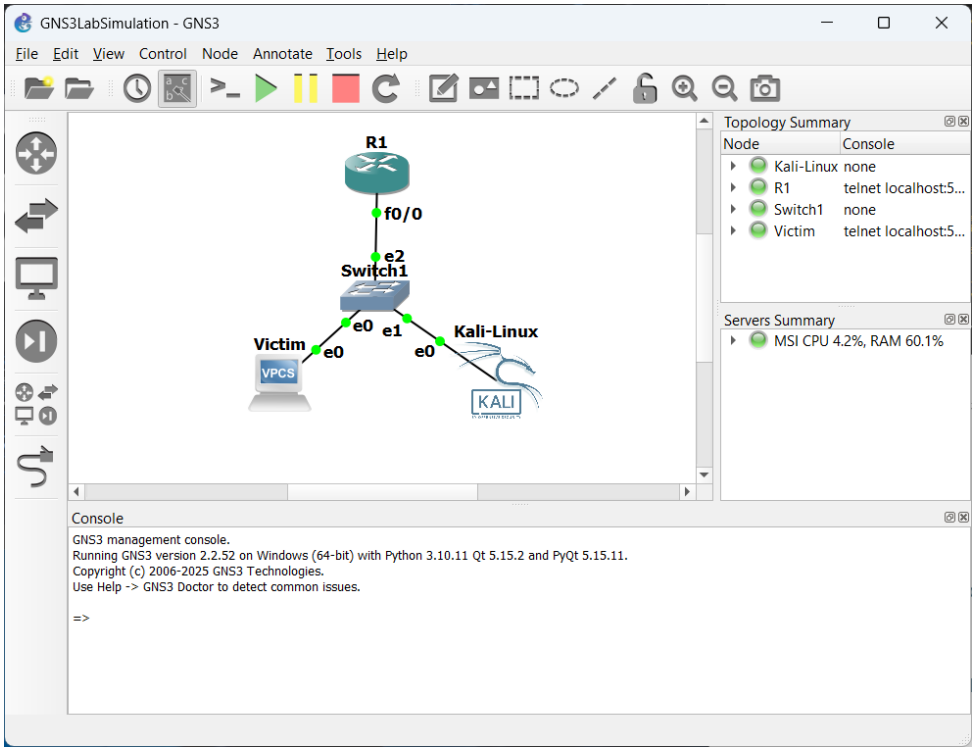
solarwinds | Solar-PuTTY *free tool* © 2019-2024 SolarWinds Worldwide, LLC. All rights reserved.

c. Set ip address for f0/0

```
R1#en
R1#config
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#int f0/0
R1(config-if)#ip add 192.168.0.1 255.255.255.0
R1(config-if)#no shut
R1(config-if)#
*Mar  1 00:03:44.847: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:03:45.847: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1(config-if)#exit
R1(config)#
```

solarwinds | Solar-PuTTY *free tool* © 2019-2024 SolarWinds Worldwide, LLC. All rights reserved.

3. On ‘Victim’ and ‘Kali-Linux’, tap ‘Start’



4. On ‘Victim’
   a. Set ip address for VPCS

```
Victim> ip 192.168.0.77/24 192.168.0.1
Checking for duplicate address...
PC1 : 192.168.0.77 255.255.255.0 gateway 192.168.0.1

Victim> show ip

NAME        : Victim[1]
IP/MASK     : 192.168.0.77/24
GATEWAY     : 192.168.0.1
DNS         :
MAC         : 00:50:79:66:68:00
LPORT       : 10010
RHOST:PORT  : 127.0.0.1:10011
MTU:        : 1500
```

b. Test connectivity with router

```
Victim> ping 192.168.0.1
84 bytes from 192.168.0.1 icmp_seq=1 ttl=255 time=8.731 ms
84 bytes from 192.168.0.1 icmp_seq=2 ttl=255 time=9.611 ms
84 bytes from 192.168.0.1 icmp_seq=3 ttl=255 time=1.983 ms
84 bytes from 192.168.0.1 icmp_seq=4 ttl=255 time=1.840 ms
84 bytes from 192.168.0.1 icmp_seq=5 ttl=255 time=2.918 ms
```

5. On ‘Kali-Linux’
   a. Assign ip subnet for kali linux

```
PS> user00@kali: /home/user00
File  Actions  Edit  View  Help
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:27:b8:a1 brd ff:ff:ff:ff:ff:ff

┌──(user00㉿kali)-[/home/user00]
└─PS> sudo ip addr add 192.168.0.80/24 dev eth0
[sudo] password for user00:

┌──(user00㉿kali)-[/home/user00]
└─PS> ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group def
ault 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: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
roup default qlen 1000
    link/ether 08:00:27:27:b8:a1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.80/24 scope global eth0
       valid_lft forever preferred_lft forever
```

   b. Assign default getaway

```
┌──(user00㉿kali)-[/home/user00]
└─PS> sudo ip link set eth0 up
[sudo] password for user00:

┌──(user00㉿kali)-[/home/user00]
└─PS> sudo ip routr add default via 192.168.0.1
Object "routr" is unknown, try "ip help".

┌──(user00㉿kali)-[/home/user00]
└─PS> sudo ip route add default via 192.168.0.1

┌──(user00㉿kali)-[/home/user00]
└─PS> sS
```

Generating traffic

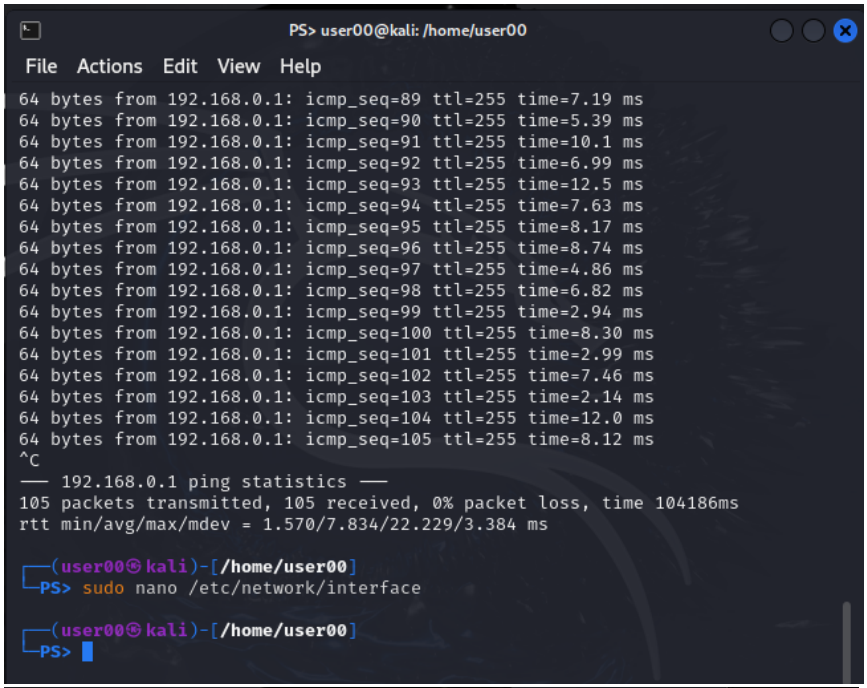
1. Ping test in kali

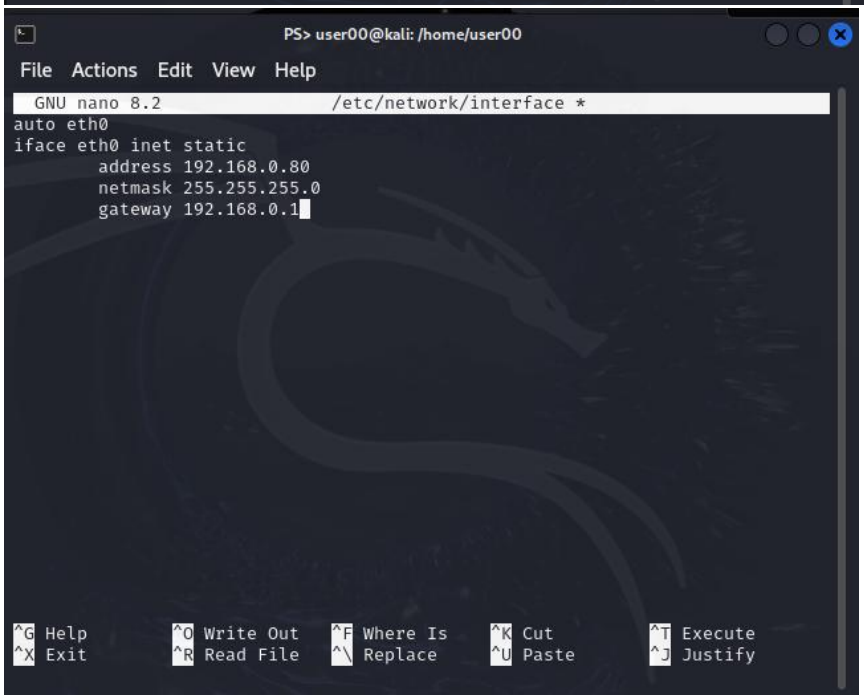
```
┌──(user00㉿kali)-[/home/user00]
└─PS> ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=255 time=22.2 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=255 time=4.52 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=255 time=10.7 ms
64 bytes from 192.168.0.1: icmp_seq=4 ttl=255 time=8.55 ms
64 bytes from 192.168.0.1: icmp_seq=5 ttl=255 time=4.19 ms
64 bytes from 192.168.0.1: icmp_seq=6 ttl=255 time=5.96 ms
64 bytes from 192.168.0.1: icmp_seq=7 ttl=255 time=10.4 ms
64 bytes from 192.168.0.1: icmp_seq=8 ttl=255 time=8.48 ms
```

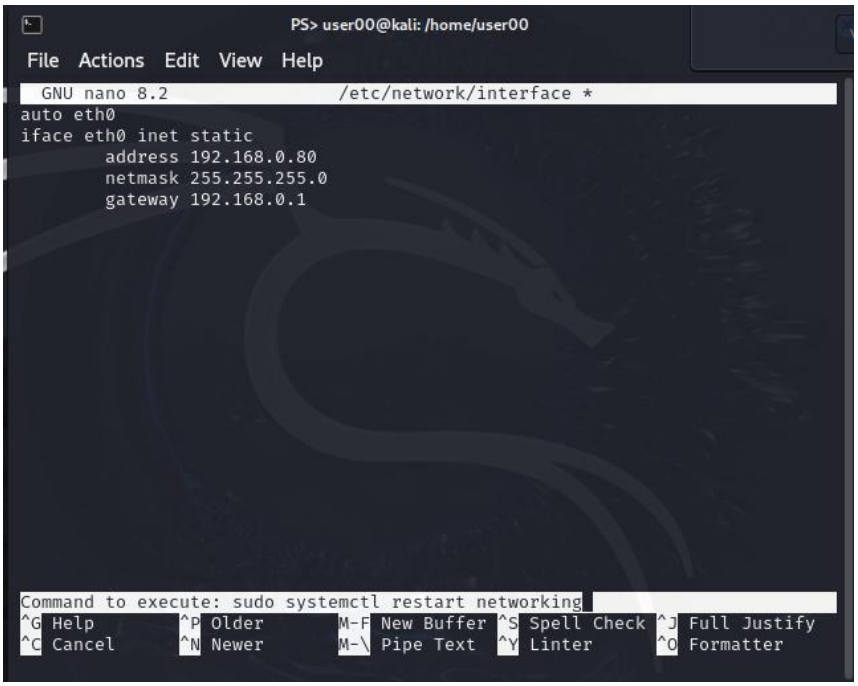
2. End ping test with Ctrl+C

```
PS> user00@kali: /home/user00
File  Actions  Edit  View  Help
64 bytes from 192.168.0.1: icmp_seq=86 ttl=255 time=4.19 ms
64 bytes from 192.168.0.1: icmp_seq=87 ttl=255 time=1.57 ms
64 bytes from 192.168.0.1: icmp_seq=88 ttl=255 time=9.42 ms
64 bytes from 192.168.0.1: icmp_seq=89 ttl=255 time=7.19 ms
64 bytes from 192.168.0.1: icmp_seq=90 ttl=255 time=5.39 ms
64 bytes from 192.168.0.1: icmp_seq=91 ttl=255 time=10.1 ms
64 bytes from 192.168.0.1: icmp_seq=92 ttl=255 time=6.99 ms
64 bytes from 192.168.0.1: icmp_seq=93 ttl=255 time=12.5 ms
64 bytes from 192.168.0.1: icmp_seq=94 ttl=255 time=7.63 ms
64 bytes from 192.168.0.1: icmp_seq=95 ttl=255 time=8.17 ms
64 bytes from 192.168.0.1: icmp_seq=96 ttl=255 time=8.74 ms
64 bytes from 192.168.0.1: icmp_seq=97 ttl=255 time=4.86 ms
64 bytes from 192.168.0.1: icmp_seq=98 ttl=255 time=6.82 ms
64 bytes from 192.168.0.1: icmp_seq=99 ttl=255 time=2.94 ms
64 bytes from 192.168.0.1: icmp_seq=100 ttl=255 time=8.30 ms
64 bytes from 192.168.0.1: icmp_seq=101 ttl=255 time=2.99 ms
64 bytes from 192.168.0.1: icmp_seq=102 ttl=255 time=7.46 ms
64 bytes from 192.168.0.1: icmp_seq=103 ttl=255 time=2.14 ms
64 bytes from 192.168.0.1: icmp_seq=104 ttl=255 time=12.0 ms
64 bytes from 192.168.0.1: icmp_seq=105 ttl=255 time=8.12 ms
^C
── 192.168.0.1 ping statistics ──
105 packets transmitted, 105 received, 0% packet loss, time 104186ms
rtt min/avg/max/mdev = 1.570/7.834/22.229/3.384 ms
```

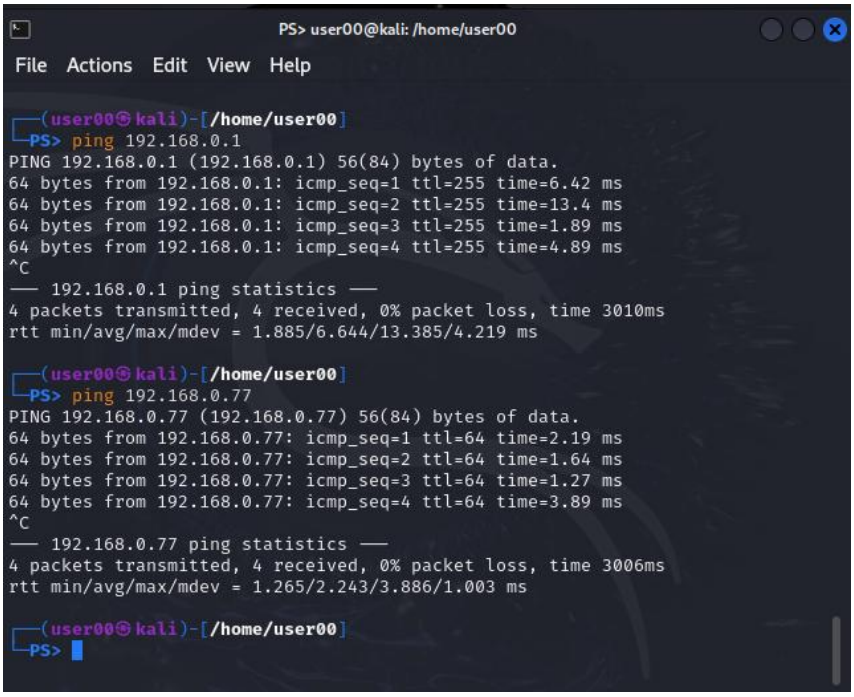
3. Modify network configuration file to ensure configurations persist after reboot


PS> user00@kali: /home/user00
File Actions Edit View Help
64 bytes from 192.168.0.1: icmp_seq=89 ttl=255 time=7.19 ms
64 bytes from 192.168.0.1: icmp_seq=90 ttl=255 time=5.39 ms
64 bytes from 192.168.0.1: icmp_seq=91 ttl=255 time=10.1 ms
64 bytes from 192.168.0.1: icmp_seq=92 ttl=255 time=6.99 ms
64 bytes from 192.168.0.1: icmp_seq=93 ttl=255 time=12.5 ms
64 bytes from 192.168.0.1: icmp_seq=94 ttl=255 time=7.63 ms
64 bytes from 192.168.0.1: icmp_seq=95 ttl=255 time=8.17 ms
64 bytes from 192.168.0.1: icmp_seq=96 ttl=255 time=8.74 ms
64 bytes from 192.168.0.1: icmp_seq=97 ttl=255 time=4.86 ms
64 bytes from 192.168.0.1: icmp_seq=98 ttl=255 time=6.82 ms
64 bytes from 192.168.0.1: icmp_seq=99 ttl=255 time=2.94 ms
64 bytes from 192.168.0.1: icmp_seq=100 ttl=255 time=8.30 ms
64 bytes from 192.168.0.1: icmp_seq=101 ttl=255 time=2.99 ms
64 bytes from 192.168.0.1: icmp_seq=102 ttl=255 time=7.46 ms
64 bytes from 192.168.0.1: icmp_seq=103 ttl=255 time=2.14 ms
64 bytes from 192.168.0.1: icmp_seq=104 ttl=255 time=12.0 ms
64 bytes from 192.168.0.1: icmp_seq=105 ttl=255 time=8.12 ms
^C
— 192.168.0.1 ping statistics —
105 packets transmitted, 105 received, 0% packet loss, time 104186ms
rtt min/avg/max/mdev = 1.570/7.834/22.229/3.384 ms
(user00㉿kali)-[/home/user00]
PS> sudo nano /etc/network/interface
(user00㉿kali)-[/home/user00]
PS>


PS> user00@kali: /home/user00
File Actions Edit View Help
GNU nano 8.2 /etc/network/interface *
auto eth0
iface eth0 inet static
address 192.168.0.80
netmask 255.255.255.0
gateway 192.168.0.1
^G Help ^O Write Out ^F Where Is ^K Cut ^T Execute
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify

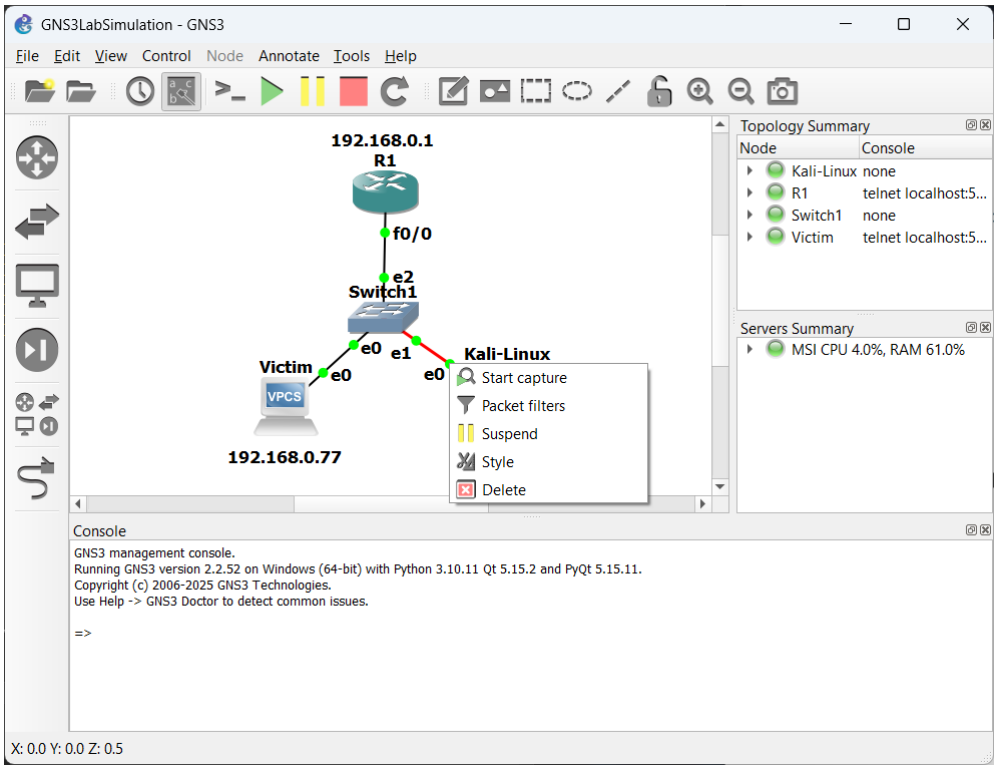


4. Sending icmp ping request over to victim to test connectivity

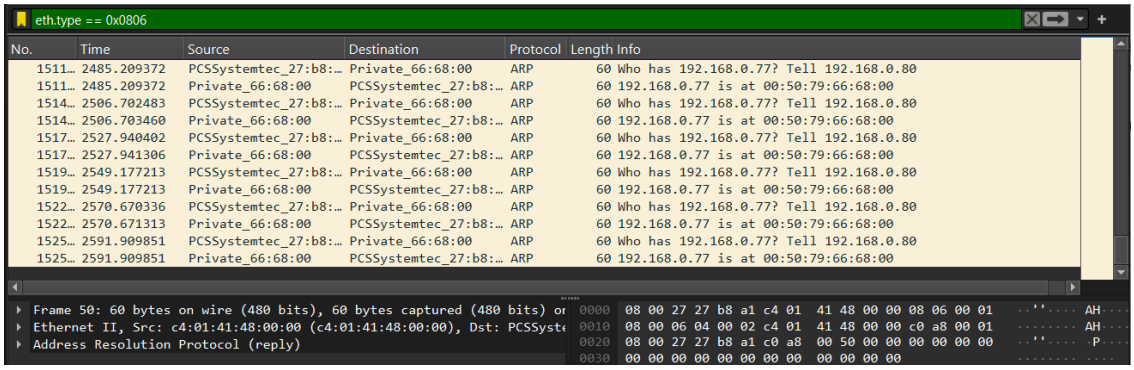


Capturing and filtering packets in Wireshark

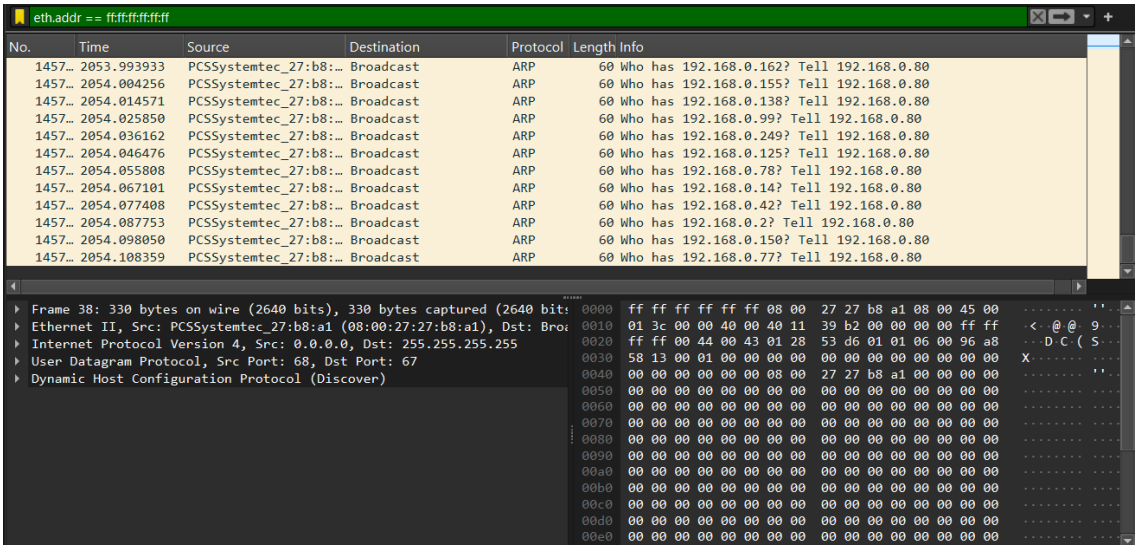
1. Start capturing packets



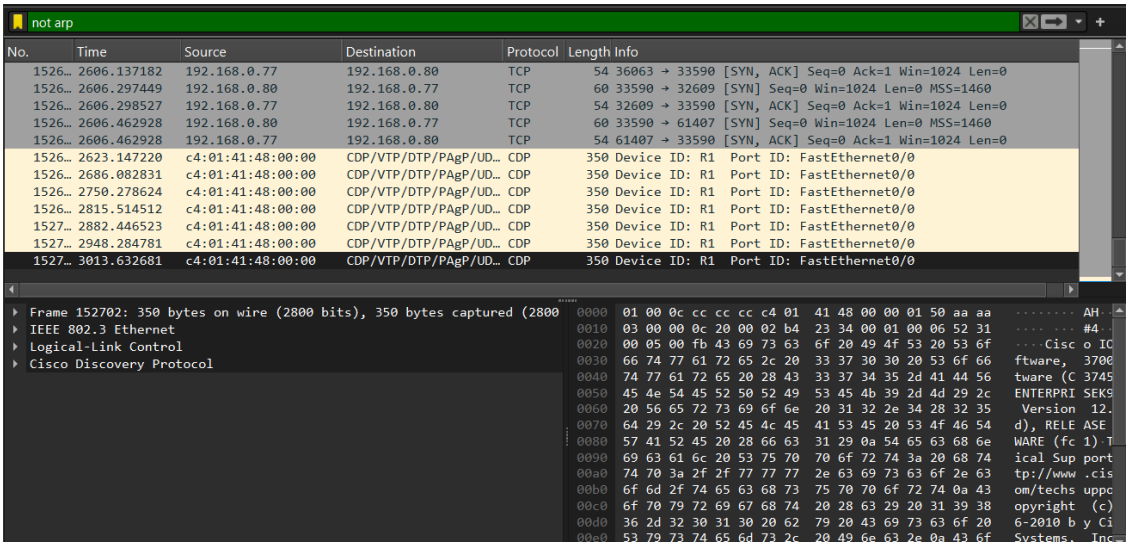
2. Ethernet type 0x0806 (ARP)
    a. Filter: eth.type == 0x0806



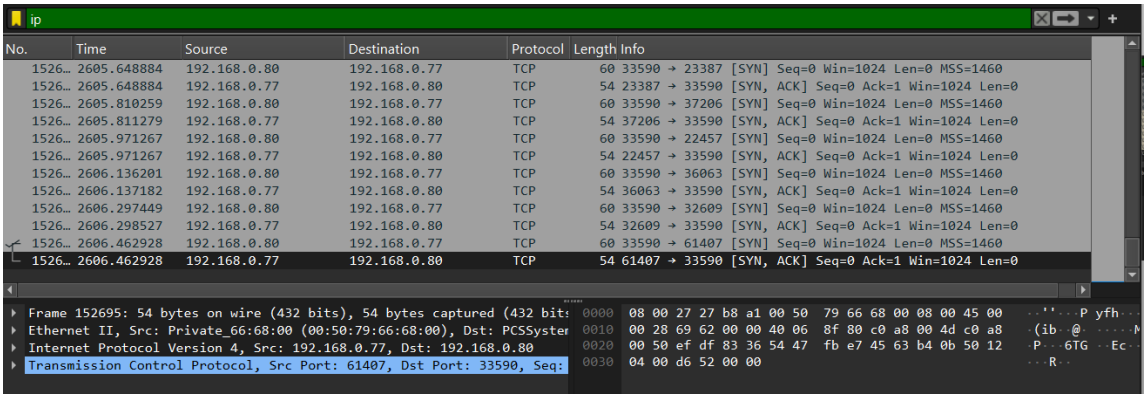
3. Ethernet broadcast
    a. Filter: eth.addr == ff:ff:ff:ff:ff:ff



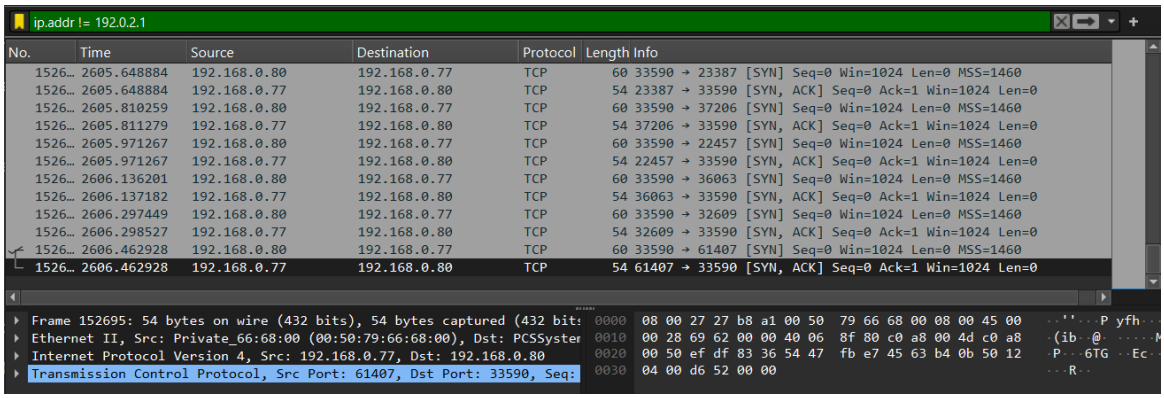
4. No ARP
    a. Filter: not arp



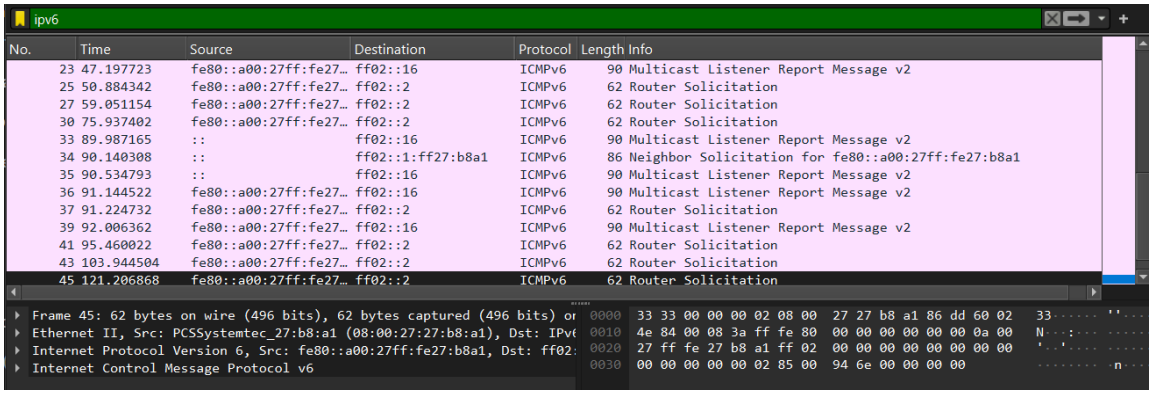
5. IPv4 only
    a. Filter: ip



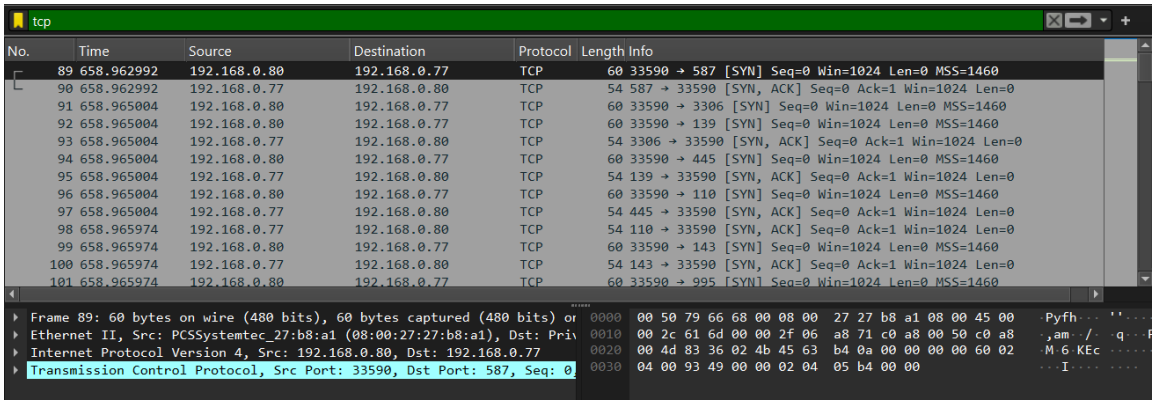
6. IPv4 address isn't 192.0.2.1
    a. Filter: ip.addr != 192.0.2.1



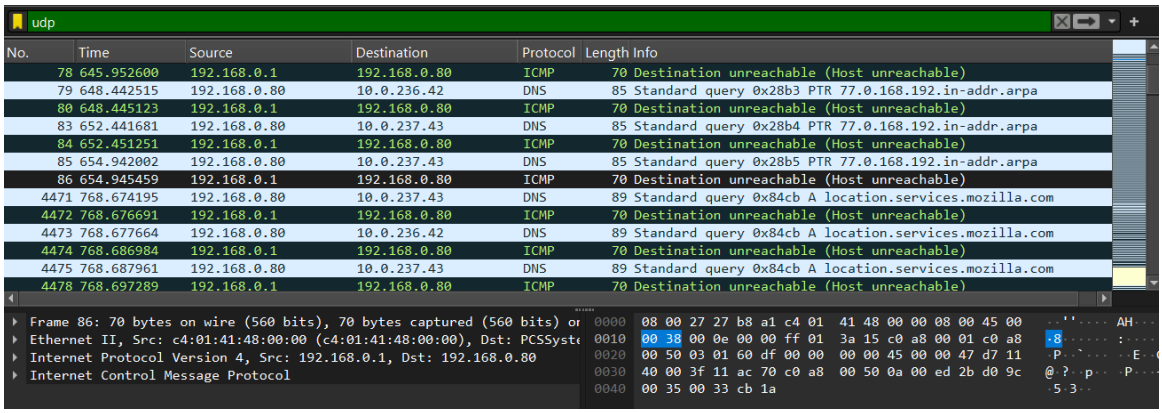
7. IPv6 only
    a. Filter: ipv6



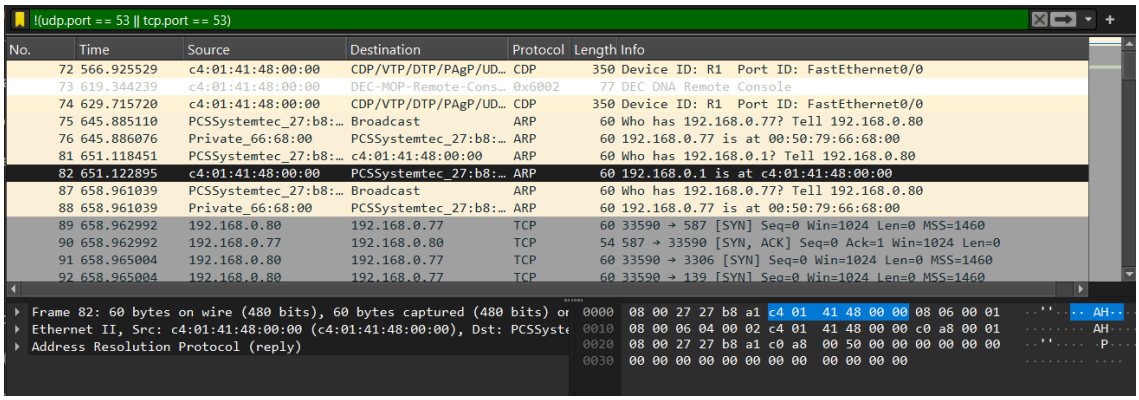
8. TCP only
    a. Filter: tcp



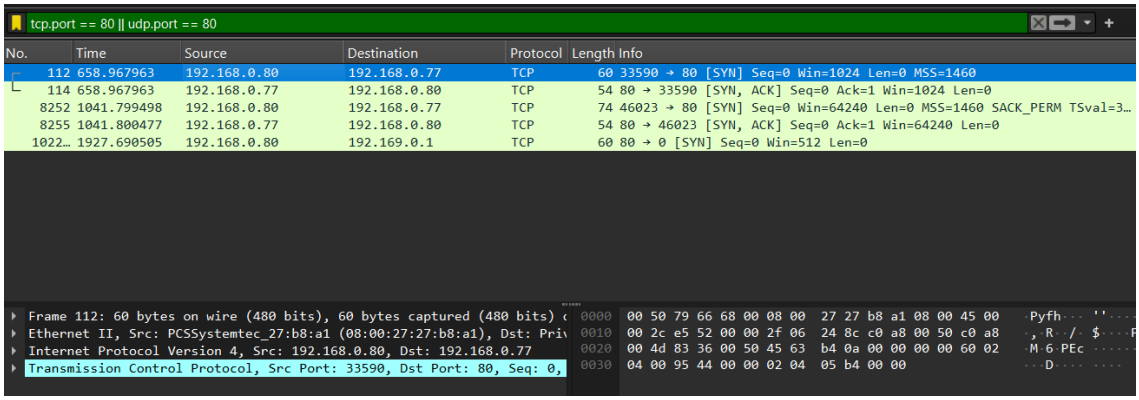
9. UDP only
    a. Filter: udp



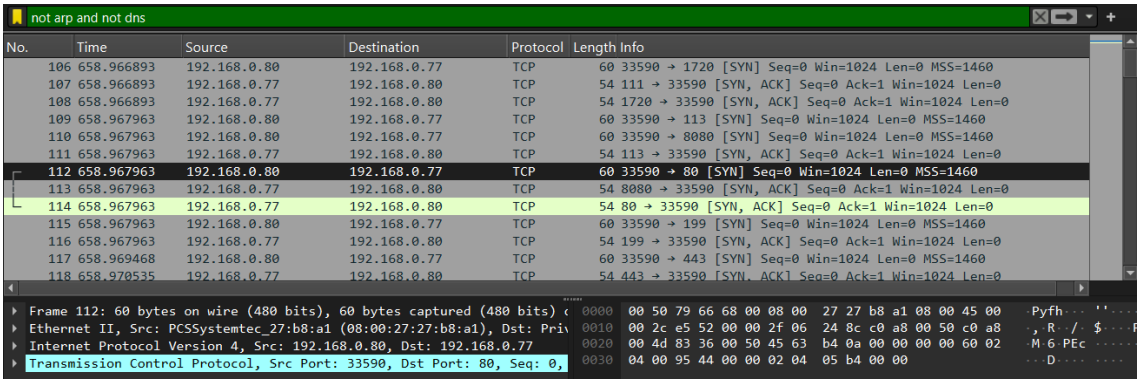
10. Non-DNS port
    a. Filter: !(udp.port == 53 || tcp.port == 53)



11. TCP or UDP port is 80 (HTTP)
    a. Filter: tcp.port == 80 || udp.port == 80

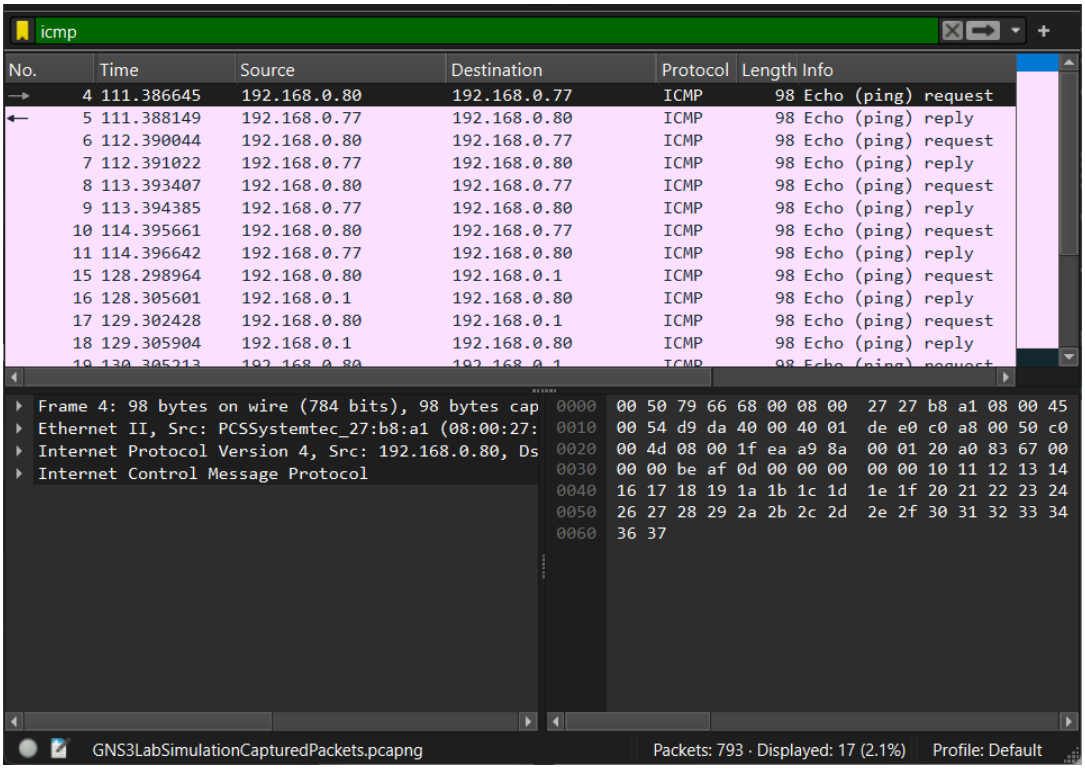


12. No ARP and no DNS
    a. Filter: not arp and not dns



13. ICMP

a. Filter: icmp



Analyzing the captured packets for security threats

1. Ethernet type 0x0806 (ARP)
    a. Filter: eth.type == 0x0806
    b. Excessive Arp traffic could indicate Arp spoofing or floods
    c. Can disrupt communication or redirect traffic malicious actors
2. Ethernet broadcast
    a. Filter: eth.addr == ff:ff:ff:ff:ff:ff
    b. Broadcast storms or flood can be indication of DoS attaks
    c. Can reveal network misconfiguration
3. No ARP
    a. Filter: not arp
    b. ARP traffic exclusion can overlook ARP-related attacks
4. IPv4 only
    a. Filter: ip
    b. Potentially missing ipv6 adverse threats
5. IPv4 address isn't 192.0.2.1
    a. Filter: ip.addr != 192.0.2.1
    b. Good to isolate other traffic
6. IPv6 only
    a. Filter: ipv6
    b. Can reveal ipv6 specific attacks
7. TCP only
    a. Filter: tcp
    b. Focus on attacks like SYN flood
8. UDP only
    a. Filter: udp
    b. Detecting DNS ampilifaction, UDP flood or other UDP-based attacks
9. Non-DNS port
    a. Filter: !(udp.port == 53 || tcp.port == 53)
    b. Identify non DNS traffic but risk overlooking DNS tunnelling attacks
10. TCP or UDP port is 80 (HTTP)
    a. Filter: tcp.port == 80 || udp.port == 80
    b. Identify httyp-based attacks like htyp flood or malicious payload
11. No ARP and no DNS
    a. Filter: not arp and not dns
    b. Missing ARP spoofing or DNS tunneling attack
12. ICMP
    a. Filter: icmp
    b. Detect icmp -based attacks like ping flood or reconnaissance