Robert Carpenter
github.com/robertmcarpenter
April 4th 2025

Setting up my own “mock” Enterprise vLab Environment - Step 1 - Understanding the LibVirt Virtual Network before installing Ubuntu Server

A Documentary of Steps of building a Virtual Isolated Lab for Cyber Security Research on QEMU/KVM under a Kali Host

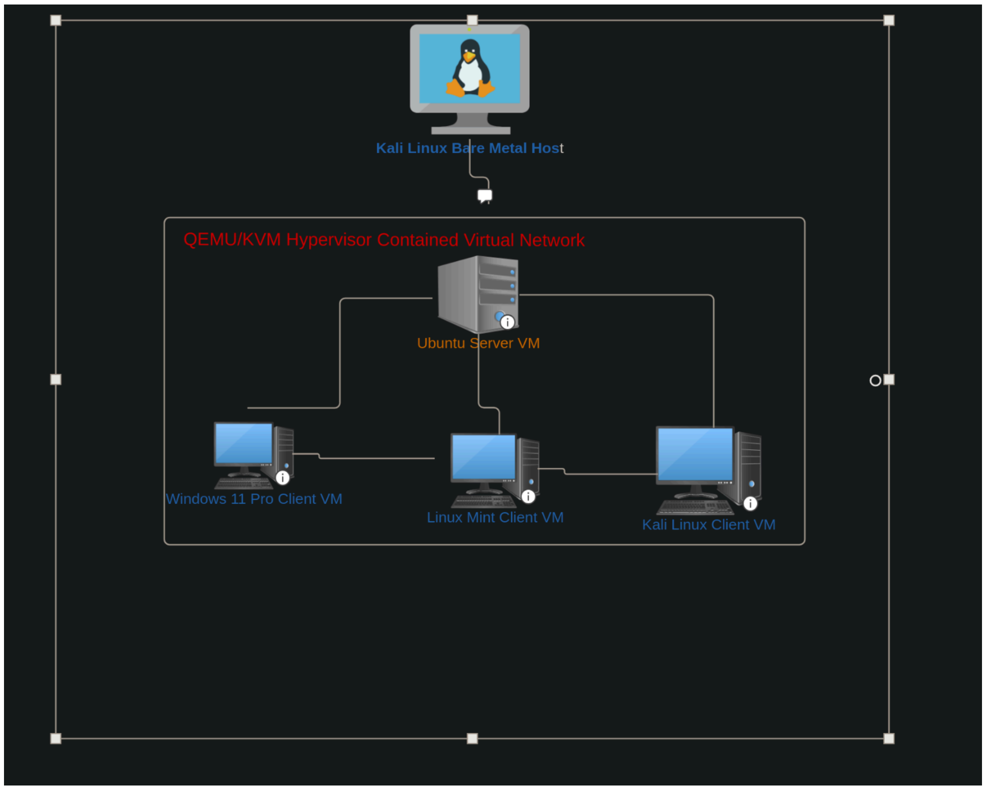
Lab Goal: To explain and demonstrate the Network Topology of the virtual Lab network I am creating which is self contained within a QEMU/KVM Virtual network running on my Linux Host. This is to establish an understanding of how Network Traffic will be routed in this Virtual Network environment , and how Guest Vms will interact and “talk” to other guests or even other devices on the physical LAN.

Topics Covered: ARP Requests, Layer 2 versus Layer 3, NAT, libvirt , IP Forwarding, LAN, DHCP, DNS, Linux Commands, Windows Commands, QEMU/KVM Virtualization Hypervisor for Linux, Private IP Ranges (Class A/B/C)

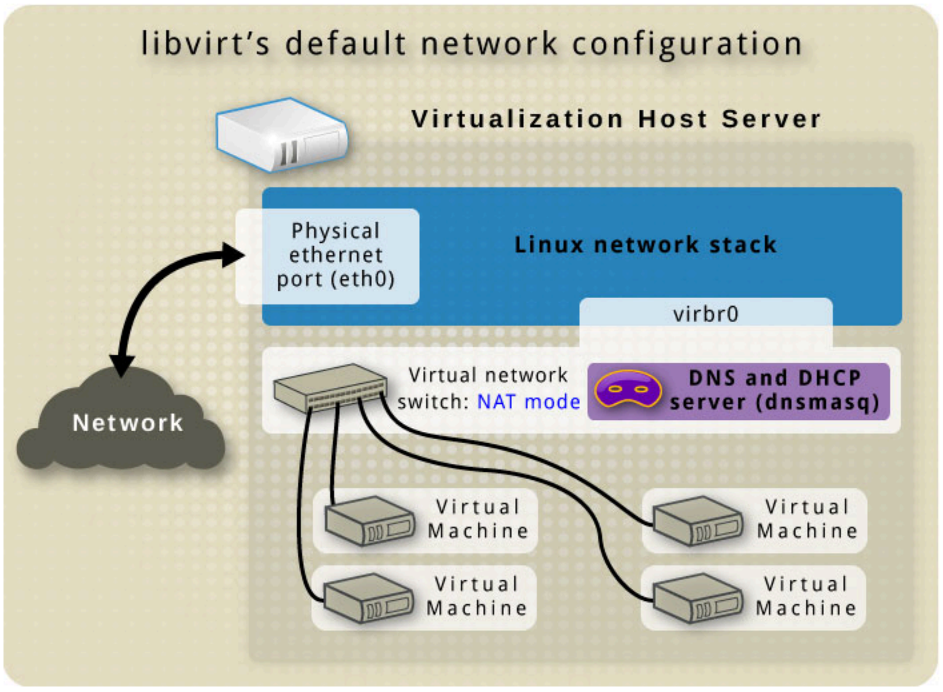
What you need to follow along:

- A PC with at least 16 GB RAM and CPU that supports Virtualization
- Download the following (on your Debian based Linux host)
  - Ubuntu Server .iso image
  - Windows 11 24H2 .iso image
  - Install libvirt , qemu, and virt-manager packages.
- Wireshark (on your VM guests to do packet inspection)

Robert Carpenter
github.com/robertmcarpenter
April 4th 2025



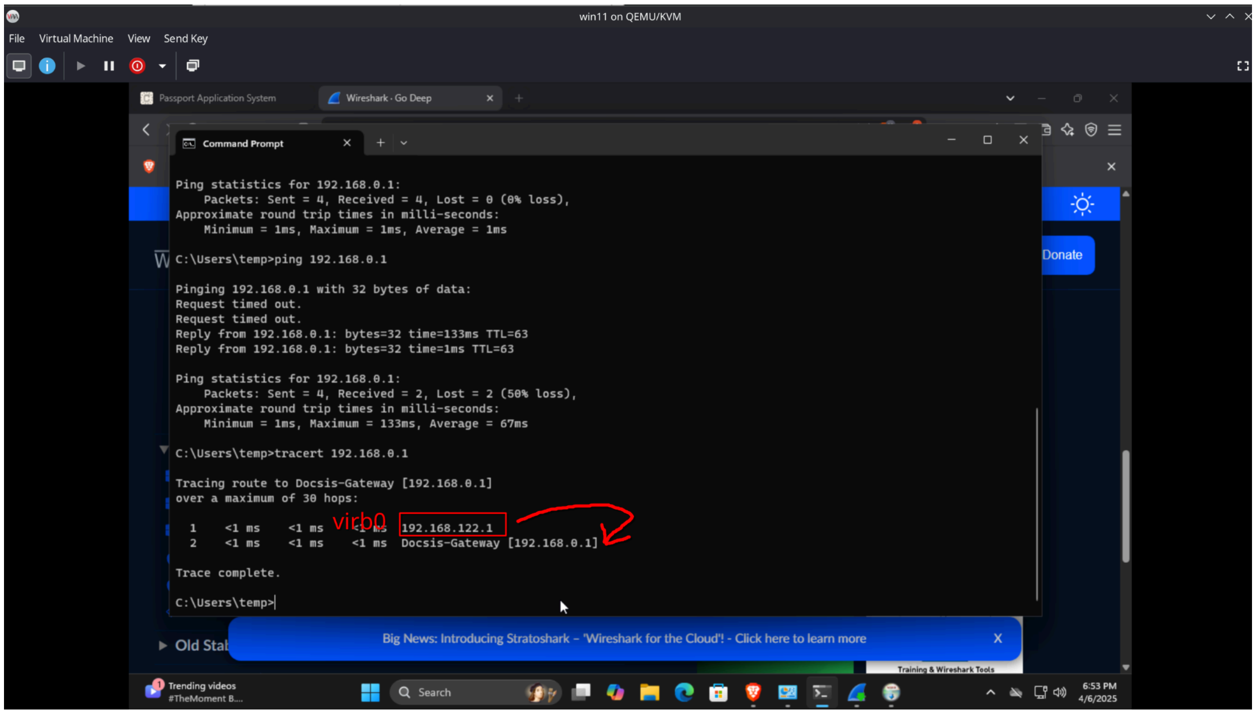
This next picture describes the virtual network that is setup by default with **NAT** on libvirt, the daemon that manages virtualization in Linux. This is taken from the wiki for libvirt.


libvirt's default network configuration
Virtualization Host Server
Physical ethernet port (eth0)
Linux network stack
virbr0
Network
Virtual network switch: NAT mode
DNS and DHCP server (dnsmasq)
Virtual Machine
Virtual Machine
Virtual Machine
Virtual Machine

Robert Carpenter
github.com/robertmcarpenter
April 4th 2025

As you can see here , libvirt has a few virtual network devices that it creates. The first device is called virbr0 which is a bridge to your real network interface. If I were on a guest VM inside the network pictured above, the ICMP packet would travel to the virb0 interface first , which serves as the gateway for the internal VM network. Once it hits the virb0 interface, the packet will then travel to my real ethernet adapter via a “bridge” which is eth0. I can demonstrate this by issuing a **traceroute** command to **192.168.0.1** (my real physical LAN router/gateway to the internet provided by my ISP) and see how the packet travels from the Virtual Network outside to my physical one. I will also try to ping my physical LAN gateway to see if I can reach it from within the VM.

Now that I am in my Guest Windows 11 Pro VM on QEMU/KVM (note again this is managed by libvirt) I will issue the command: **tracert 192.168.0.1**

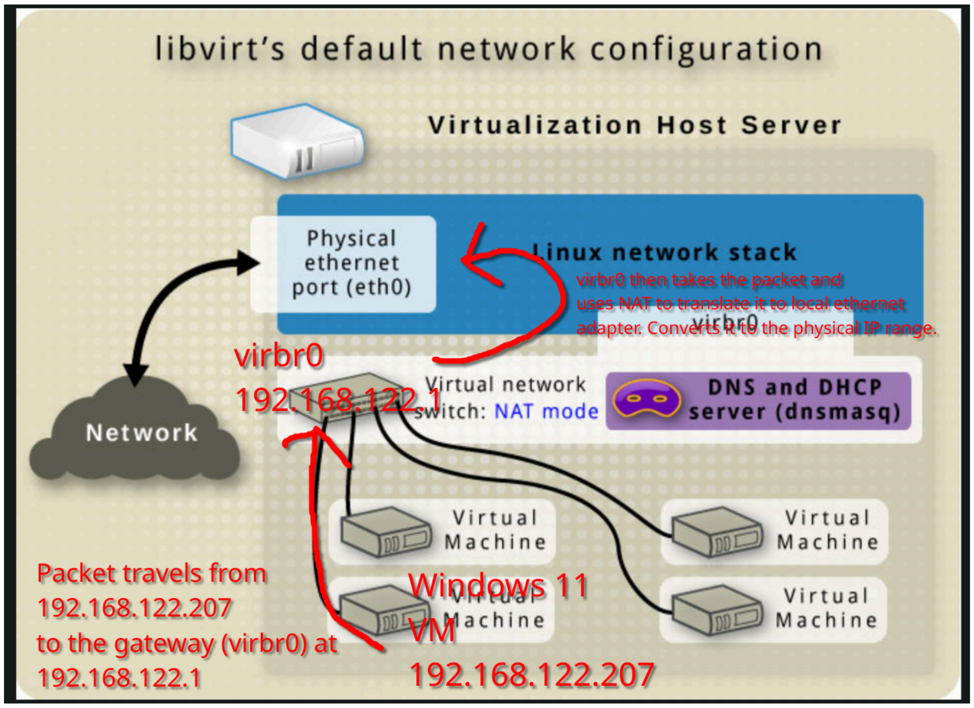


```
Ping statistics for 192.168.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms

C:\Users\temp>ping 192.168.0.1

Pinging 192.168.0.1 with 32 bytes of data:
Request timed out.
Request timed out.
Reply from 192.168.0.1: bytes=32 time=133ms TTL=63
Reply from 192.168.0.1: bytes=32 time=1ms TTL=63

Ping statistics for 192.168.0.1:
    Packets: Sent = 4, Received = 2, Lost = 2 (50% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 133ms, Average = 67ms

C:\Users\temp>tracert 192.168.0.1

Tracing route to Docsis-Gateway [192.168.0.1]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  192.168.122.1
  2    <1 ms    <1 ms    <1 ms  Docsis-Gateway [192.168.0.1]

Trace complete.

C:\Users\temp>
```

Notice how the ping failed? How is that when I ran traceroute, the ping failed but traceroute was able to resolve 192.168.0.1 if it’s supposedly unreachable? **This is because of NAT (Network Address Translation).** My Windows 11 Guest VM doesn’t even know that **192.168.0.1** even exists. The IP address is changed/mapped to the real one by the internal network. My Windows 11 Guest VM is only aware of the other guest virtual machine on the virtual internal NAT network as well as its “router” which is **virb0.**

Robert Carpenter
github.com/robertmcarpenter
April 4th 2025

Referring back to the diagram provided by the Libvirt Wiki we can see this in action. **Virb0** is forwarding packets from the virtual internal NAT network onto the physical one. **Virb0 ALSO acts as an all-in-one router and DHCP Server (much like the routers in your house provided by your ISP!).**


libvirt's default network configuration
Virtualization Host Server
Physical ethernet port (eth0)
Linux network stack
virbr0 then takes the packet and uses NAT to translate it to local ethernet adapter. Converts it to the physical IP range.
virbr0
virbr0
192.168.122.1
Virtual network switch: NAT mode
DNS and DHCP server (dnsmasq)
Network
Virtual Machine
Virtual Machine
Virtual Machine
Virtual Machine
Windows 11 VM
192.168.122.207
Packet travels from 192.168.122.207 to the gateway (virbr0) at 192.168.122.1

I've annotated the diagram to better reflect my explanation. When I issued that traceroute command, packet travels as such:

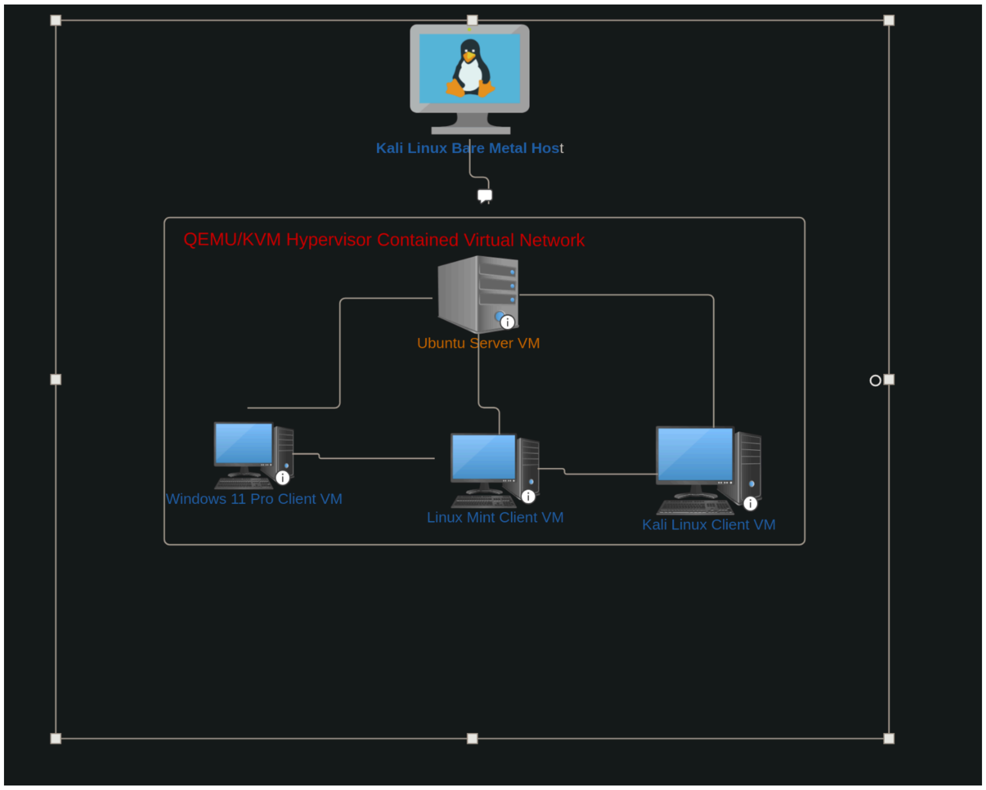
Guest VM 192.168.122.207 —-> (virbr0) 192.168.122.1 —> NAT translates 192.168.122.1 to ——-> IP address of my Physical Host 192.168.0.126.

This same technology is why we have Private IPs in our internal network. It is the job of the router on the network to effectively use 1 IP address for all hosts on a single network.

Essentially what I'm trying to say is that this internal NAT network functions exactly like the network you have at home which you have bought from your ISP. Your ISP assigns you a public IP Address from which all your devices on your network can use. The same concept applies to this Libvirt QEMU/KVM internal network.

I would like to change this configuration so that ALL TRAFFIC goes through my Ubuntu Server I want to set up.

Robert Carpenter
github.com/robertmcarpenter
April 4th 2025



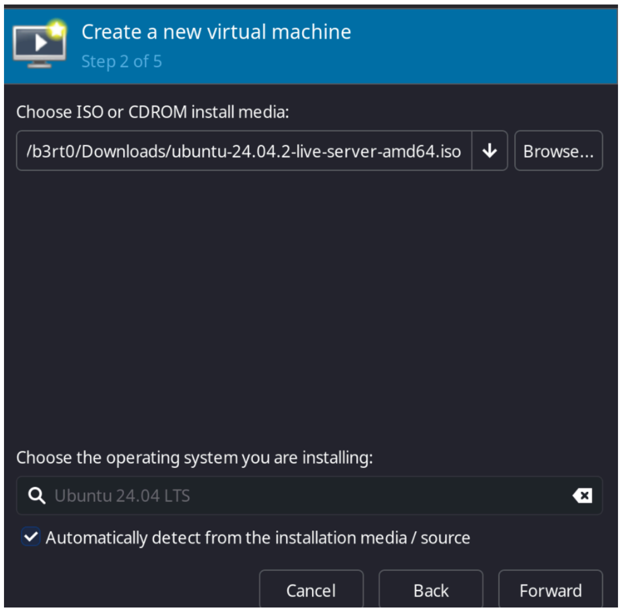
Once again here is the topology of the virtual network I want to set up. Clients on this virtual network can only access the outer world by going through my Ubuntu Server. The goal that I want to achieve with this is to treat the Ubuntu Server as a firewall/router just like in a typical corporate environment. Eventually I will set up an Active Directory Domain Controller inside the virtual network and then hook it up to the Ubuntu Server to reach the outside world or configure it in a DMZ network.

PLEASE NOTE THAT THESE INITIAL STEPS ARE TO GET THE NETWORK UP AND RUNNING FIRST! ALL NETWORK SECURITY HARDENING TECHNIQUES WILL BE ADDED LATER ONCE THIS MINI NETWORK IS PROPERLY WORKING.

Now that a baseline for how networking works in Libvirt , the virtualization API daemon that manages VMs in QEMU/KVM, let's install the Ubuntu Server VM.

I will open **virt-manager** and click the add button to create a new VM.

Robert Carpenter
github.com/robertmcarpenter
April 4th 2025

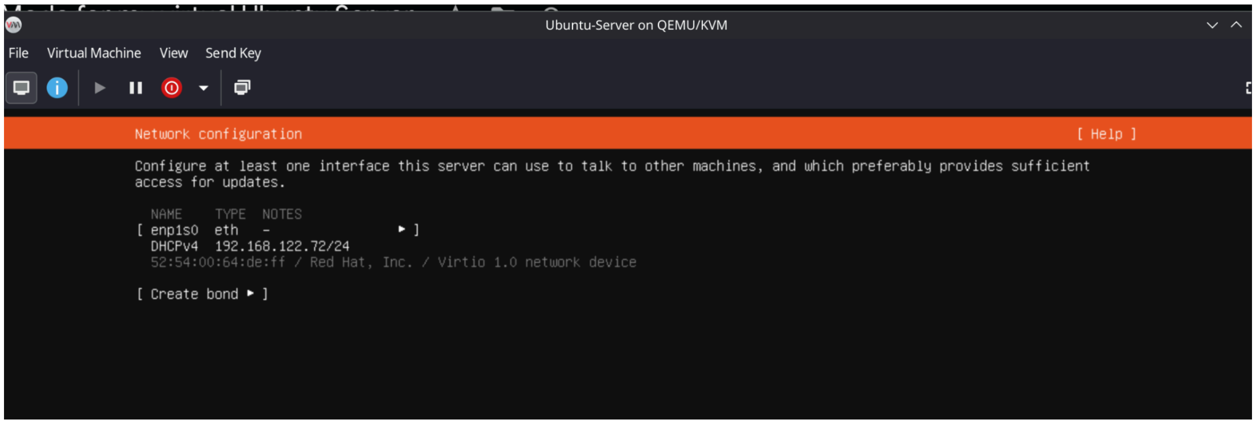


Choose ISO or CDROM install media:
/b3rt0/Downloads/ubuntu-24.04.2-live-server-amd64.iso

Choose the operating system you are installing:
Ubuntu 24.04 LTS

✓ Automatically detect from the installation media / source

I'll point virt-manager to the .ISO I downloaded from the Internet.
I'll give this VM 4 GB of RAM and a small 28GB disk size. Now, I'll finish the configuration options and setting the Device types for storage and network to use "virtio" Virtio is a special virtualized device that is KVM accelerated, essentially the device is "aware" it is in a virtual environment and reaches out directly to the host for resources instead of emulating it. Type to boot it up! Once I get to the Network configuration notice what the IP address is of this VM:



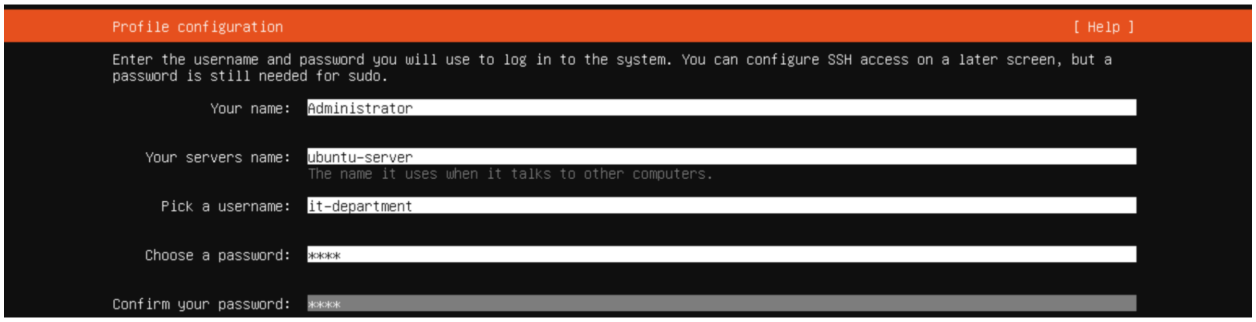
```
Network configuration                                                          [ Help ]

Configure at least one interface this server can use to talk to other machines, and which preferably provides sufficient
access for updates.

  NAME    TYPE  NOTES
[ enp1s0  eth   -                ► ]
  DHCPv4  192.168.122.72/24
  52:54:00:64:de:ff / Red Hat, Inc. / Virtio 1.0 network device

[ Create bond ► ]
```

This VM contacted virbr0 which is running DHCP and DNSMasq server to receive an IP address.

Ubuntu Server (configured with DHCP) 192.168.122.72 ——-> (virbr0) 192.168.122.1

KEEP IN MIND THIS IS THE DEFAULT NAT CONFIGURATION. WE WILL CHANGE THIS LATER SO THIS VM ACTS AS OUR ROUTER AND DHCP/DNS SERVER.

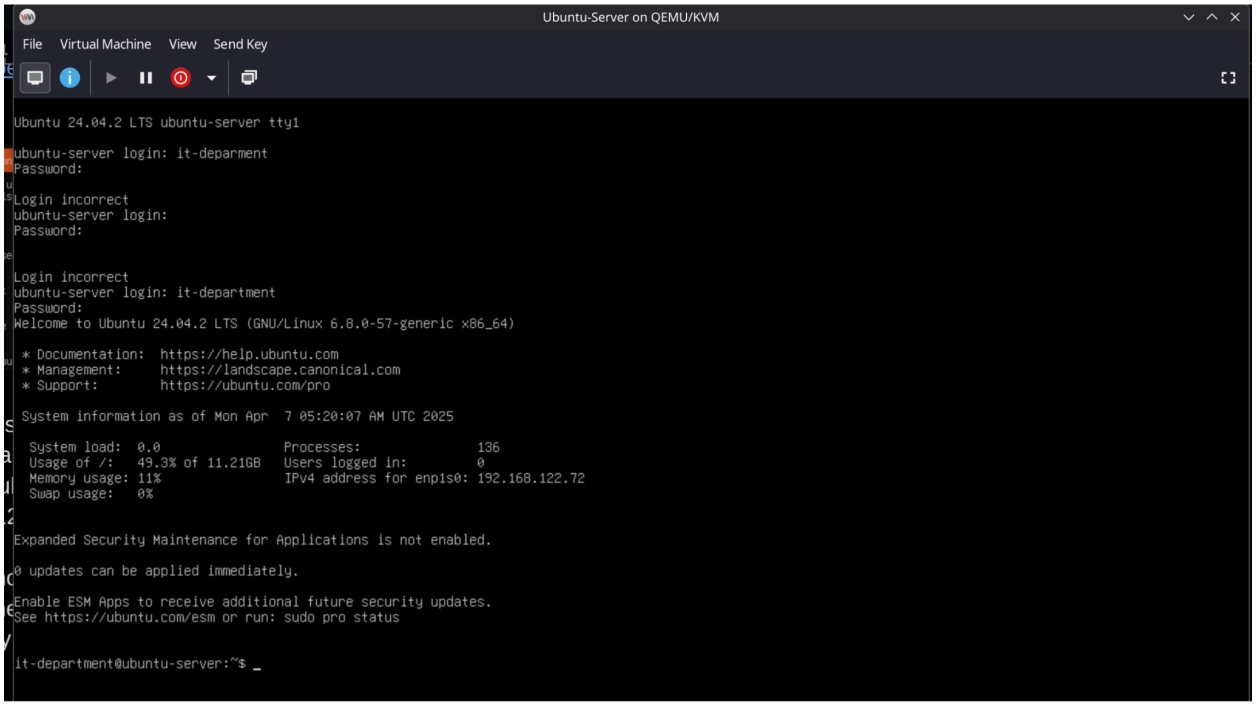
Robert Carpenter
github.com/robertmcarpenter
April 4th 2025



```
Profile configuration                                                                    [ Help ]

Enter the username and password you will use to log in to the system. You can configure SSH access on a later screen, but a
password is still needed for sudo.

             Your name:  Administrator

     Your servers name:  ubuntu-server
                         The name it uses when it talks to other computers.

       Pick a username:  it-department

     Choose a password:  ****

 Confirm your password:  ****
```

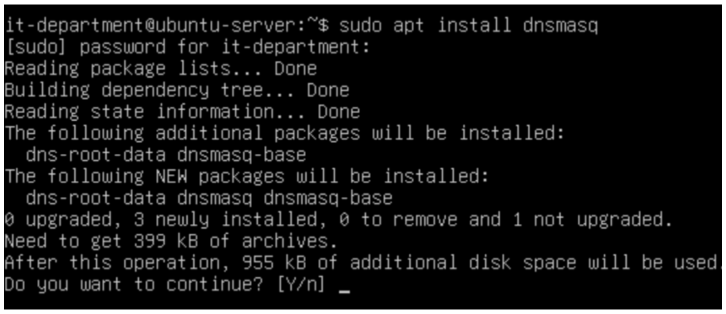
As shown these are the credentials I've set for my server.
User = it-department
Hostname = ubuntu-server
Password = 1234 (intentionally weak for later labs on password cracking)

I've mostly chosen default configurations to simply get this server up and running. I need to install all the packages for the base system and for that I need an internet connection. I will make my configuration changes later to set this up as a router and DHCP/DNS server. Once the system is done installing, I will go ahead and reboot it to check everything was installed.



```
Ubuntu 24.04.2 LTS ubuntu-server tty1

ubuntu-server login: it-deparment
Password:

Login incorrect
ubuntu-server login:
Password:


Login incorrect
ubuntu-server login: it-department
Password:
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-57-generic x86_64)

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

 System information as of Mon Apr  7 05:20:07 AM UTC 2025

  System load:  0.0               Processes:               136
  Usage of /:   49.3% of 11.21GB  Users logged in:         0
  Memory usage: 11%               IPv4 address for enp1s0: 192.168.122.72
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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


it-department@ubuntu-server:~$ _
```

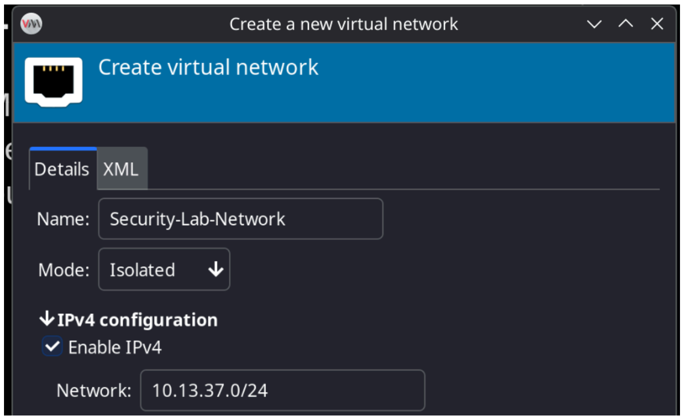
Robert Carpenter
github.com/robertmcarpenter
April 4th 2025
Success! The Ubuntu Server is now installed! Since I want this VM to act as a router and gateway to all the VMs connected underneath it I will need to install a DHCP and DNS server here. I will install **dnsmasq** a Linux package that will help me do just this.



```
it-department@ubuntu-server:~$ sudo apt install dnsmasq
[sudo] password for it-department:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  dns-root-data dnsmasq-base
The following NEW packages will be installed:
  dns-root-data dnsmasq dnsmasq-base
0 upgraded, 3 newly installed, 0 to remove and 1 not upgraded.
Need to get 399 kB of archives.
After this operation, 955 kB of additional disk space will be used
Do you want to continue? [Y/n] _
```

I will hit yes! Now that the virtual machine has dnsmasq installed, I will shut it off and open virt-manager to set up a new Virtual network. This is necessary because Virt-manager provides all the connections between the VMS, it is responsible for providing all the virtual ethernet cables and switches to “connect” the VMs together.

On Virt-Manager I will click **Edit > Check Connection Details.**



Name: Security-Lab-Network
Mode: Isolated
✓ Enable IPv4
Network: 10.13.37.0/24

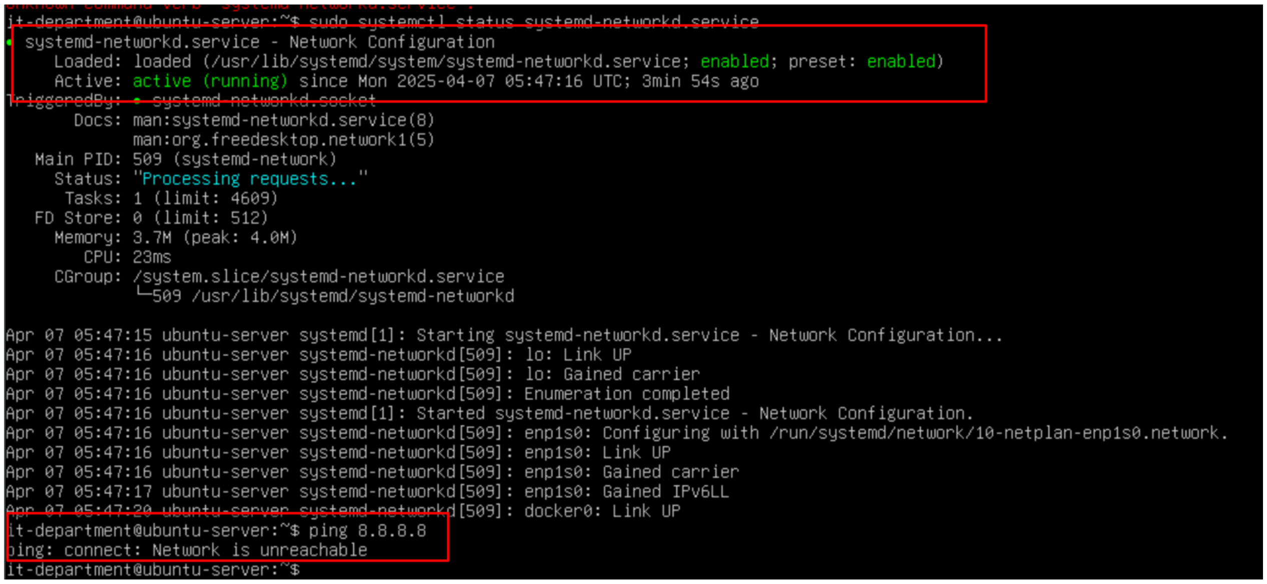
I will name this Internal Network **Security-Lab-Network**. This Network will operate as a /24 subnet so all IPs will share the first 3 octets 10.13.37.XXX but have a different ending. I also need to turn off DHCPv4 so the dnsmasq that is used by virt-manager isn’t the one that is handing out Ips. I want that job to go to my Ubuntu Server VM.

Whichever VMs I choose to connect to this network will be able to talk to each other but not outside the network. They would need a router/gateway in order to do that. That’s where our Ubuntu Server would come in! For now we need to create a network.

With the Network created I will now attach my Windows 11 Pro Vm to this Network to confirm I can ping and talk to this Ubuntu Server VM.

Robert Carpenter
github.com/robertmcarpenter
April 4th 2025

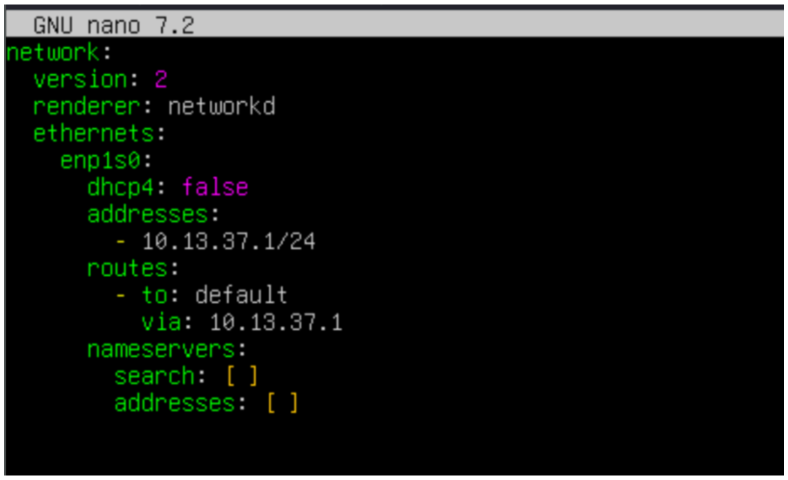
I will connect my Ubuntu Server VM to this isolated internal **Security-Lab-Network** and verify that I cannot reach out to Google using **ping 8.8.8.8**



```
it-department@ubuntu-server:~$ sudo systemctl status systemd-networkd.service
● systemd-networkd.service - Network Configuration
     Loaded: loaded (/usr/lib/systemd/system/systemd-networkd.service; enabled; preset: enabled)
     Active: active (running) since Mon 2025-04-07 05:47:16 UTC; 3min 54s ago
TriggeredBy: ● systemd-networkd.socket
       Docs: man:systemd-networkd.service(8)
             man:org.freedesktop.network1(5)
   Main PID: 509 (systemd-network)
     Status: "Processing requests..."
      Tasks: 1 (limit: 4609)
   FD Store: 0 (limit: 512)
     Memory: 3.7M (peak: 4.0M)
        CPU: 23ms
     CGroup: /system.slice/systemd-networkd.service
             └─509 /usr/lib/systemd/systemd-networkd

Apr 07 05:47:15 ubuntu-server systemd[1]: Starting systemd-networkd.service - Network Configuration...
Apr 07 05:47:16 ubuntu-server systemd-networkd[509]: lo: Link UP
Apr 07 05:47:16 ubuntu-server systemd-networkd[509]: lo: Gained carrier
Apr 07 05:47:16 ubuntu-server systemd-networkd[509]: Enumeration completed
Apr 07 05:47:16 ubuntu-server systemd[1]: Started systemd-networkd.service - Network Configuration.
Apr 07 05:47:16 ubuntu-server systemd-networkd[509]: enp1s0: Configuring with /run/systemd/network/10-netplan-enp1s0.network.
Apr 07 05:47:16 ubuntu-server systemd-networkd[509]: enp1s0: Link UP
Apr 07 05:47:16 ubuntu-server systemd-networkd[509]: enp1s0: Gained carrier
Apr 07 05:47:17 ubuntu-server systemd-networkd[509]: enp1s0: Gained IPv6LL
Apr 07 05:47:20 ubuntu-server systemd-networkd[509]: docker0: Link UP
it-department@ubuntu-server:~$ ping 8.8.8.8
ping: connect: Network is unreachable
it-department@ubuntu-server:~$
```

Notice how I also checked to see if **systemd-networkd.service** was running. This is to ensure that the daemon service that is responsible within Ubuntu Server is running and that my ping failing wasn't a result of the network service not running!

Since DHCP isn't running in this isolated network , none of these machines that are connected have an IP Address. I will need to manually set an Ip address for my Ubuntu Server. Ubuntu Server uses netplan to set IP configurations. I will issue **sudo nano /etc/netplan/00-config-file.yaml** to manually edit and set the IP address information.

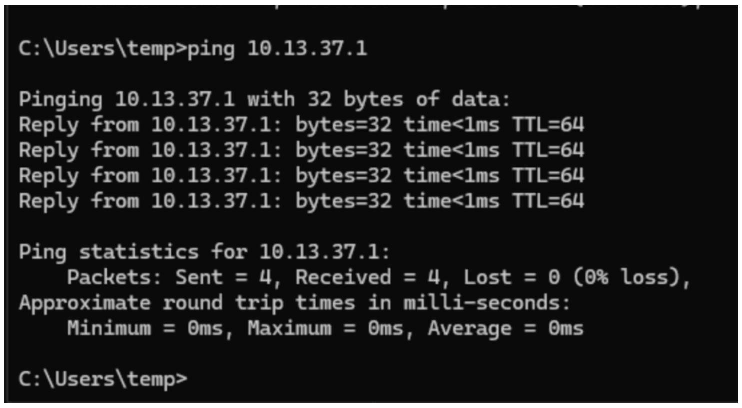


```
  GNU nano 7.2
network:
  version: 2
  renderer: networkd
  ethernets:
    enp1s0:
      dhcp4: false
      addresses:
        - 10.13.37.1/24
      routes:
        - to: default
          via: 10.13.37.1
      nameservers:
        search: [ ]
        addresses: [ ]
```

Robert Carpenter
github.com/robertmcarpenter
April 4th 2025

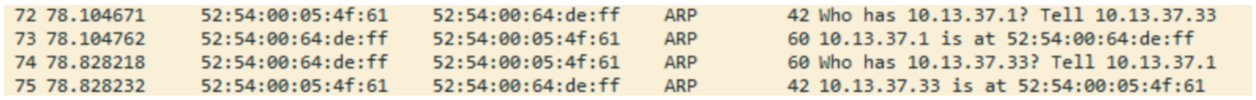
I am not sure yet what to put for the nameservers because I haven't configured dnsmasq yet. I will leave those blank for now. Once this has all been set I will issue the command: **sudo netplan apply**

Now I will check the Windows 11 Pro VM which I also have connected to the Internal Network.



```
C:\Users\temp>ping 10.13.37.1

Pinging 10.13.37.1 with 32 bytes of data:
Reply from 10.13.37.1: bytes=32 time<1ms TTL=64
Reply from 10.13.37.1: bytes=32 time<1ms TTL=64
Reply from 10.13.37.1: bytes=32 time<1ms TTL=64
Reply from 10.13.37.1: bytes=32 time<1ms TTL=64

Ping statistics for 10.13.37.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\temp>
```

If I open up Wireshark on my Windows 11 Pro VM I can see the ARP packets being sent from my Window 11 VM to my Ubuntu Server VM.

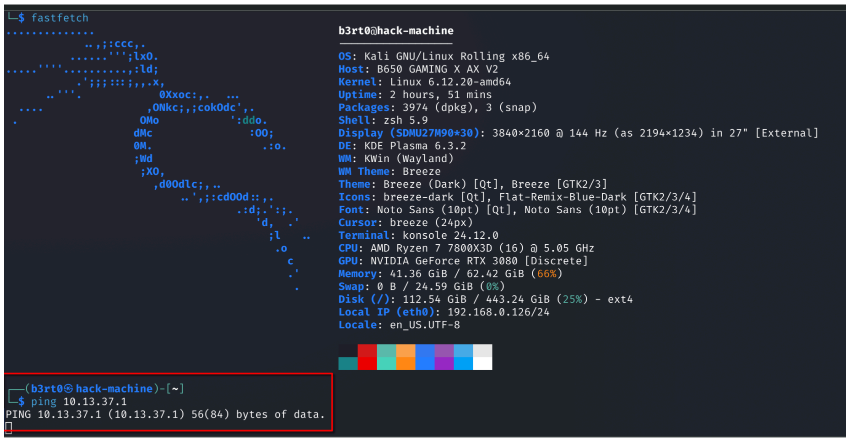


```
72 78.104671   52:54:00:05:4f:61   52:54:00:64:de:ff   ARP   42 Who has 10.13.37.1? Tell 10.13.37.33
73 78.104762   52:54:00:64:de:ff   52:54:00:05:4f:61   ARP   60 10.13.37.1 is at 52:54:00:64:de:ff
74 78.828218   52:54:00:64:de:ff   52:54:00:05:4f:61   ARP   60 Who has 10.13.37.33? Tell 10.13.37.1
75 78.828232   52:54:00:05:4f:61   52:54:00:64:de:ff   ARP   42 10.13.37.33 is at 52:54:00:05:4f:61
```

My 2 VMs have successfully negotiated each other's MAC Address and can send frames over Layer 2! They won't be able to communicate via Layer 3 , I would need to do more network configuration for this. This current setup is equivalent to having these 2 VMs connected to Physical Layer 2 Switch. A Layer 2 switch is often called a “dumb” switch because it doesn't know how to route packets outside the LAN.

To verify that these VMs are on a isolated network and have no Layer 3 Routing capabilities I will try to ping my Ubuntu Server VM from my Kali Linux Host.

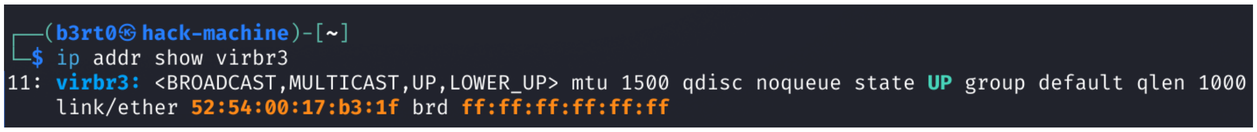
Robert Carpenter
github.com/robertmcarpenter
April 4th 2025



```
└─$ fastfetch
                                          b3rt0@hack-machine
                                          ──────────────────
                                          OS: Kali GNU/Linux Rolling x86_64
                                          Host: B650 GAMING X AX V2
                                          Kernel: Linux 6.12.20-amd64
                                          Uptime: 2 hours, 51 mins
                                          Packages: 3974 (dpkg), 3 (snap)
                                          Shell: zsh 5.9
                                          Display (SDMU27M90*30): 3840×2160 @ 144 Hz (as 2194×1234) in 27" [External]
                                          DE: KDE Plasma 6.3.2
                                          WM: KWin (Wayland)
                                          WM Theme: Breeze
                                          Theme: Breeze (Dark) [Qt], Breeze [GTK2/3]
                                          Icons: breeze-dark [Qt], Flat-Remix-Blue-Dark [GTK2/3/4]
                                          Font: Noto Sans (10pt) [Qt], Noto Sans (10pt) [GTK2/3/4]
                                          Cursor: breeze (24px)
                                          Terminal: konsole 24.12.0
                                          CPU: AMD Ryzen 7 7800X3D (16) @ 5.05 GHz
                                          GPU: NVIDIA GeForce RTX 3080 [Discrete]
                                          Memory: 41.36 GiB / 62.42 GiB (66%)
                                          Swap: 0 B / 24.59 GiB (0%)
                                          Disk (/): 112.54 GiB / 443.24 GiB (25%) - ext4
                                          Local IP (eth0): 192.168.0.126/24
                                          Locale: en_US.UTF-8

┌──(b3rt0㉿hack-machine)-[~]
└─$ ping 10.13.37.1
PING 10.13.37.1 (10.13.37.1) 56(84) bytes of data.
```

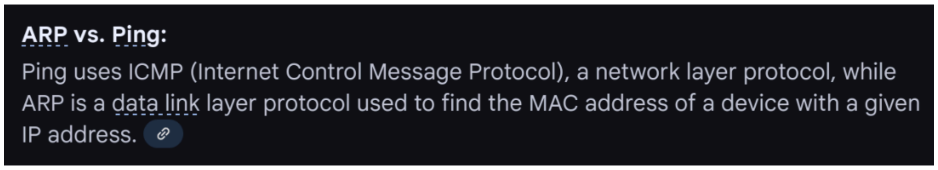
No responses came back. This could also be the case of NAT not translating from 192.168.x.x to 10.13.37.x but since my isolated network has no NAT capabilities this isn't the case.

Since these 2 VMs are connected to a “Dummy Switch” this means that if I try to look up the IP address of the interface virbr3 on my host, I shouldn’t see any IP address associated with it. Recall that my isolated network “Security-Lab-Network” is connected via the virbr3 interface on my host. Virbr3 is acting as a Layer 2 switch , this is why I was unable to reach inside this isolated network from my host. Virbr3 doesn’t know how to route packets to my host.



```
┌──(b3rt0㉿hack-machine)-[~]
└─$ ip addr show virbr3
11: virbr3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 52:54:00:17:b3:1f brd ff:ff:ff:ff:ff:ff
```

As you can see above, there is no IP Address (Layer 3) for this device. This is why when I typed **ping 10.13.37.1** the ICMP packets had no idea where to go. ICMP Packets travel on Layer 3 but my isolated internal network only has Layer 2 capabilities. **This is the key difference between Layer 2 (data layer) and Layer 3 (network layer)**.

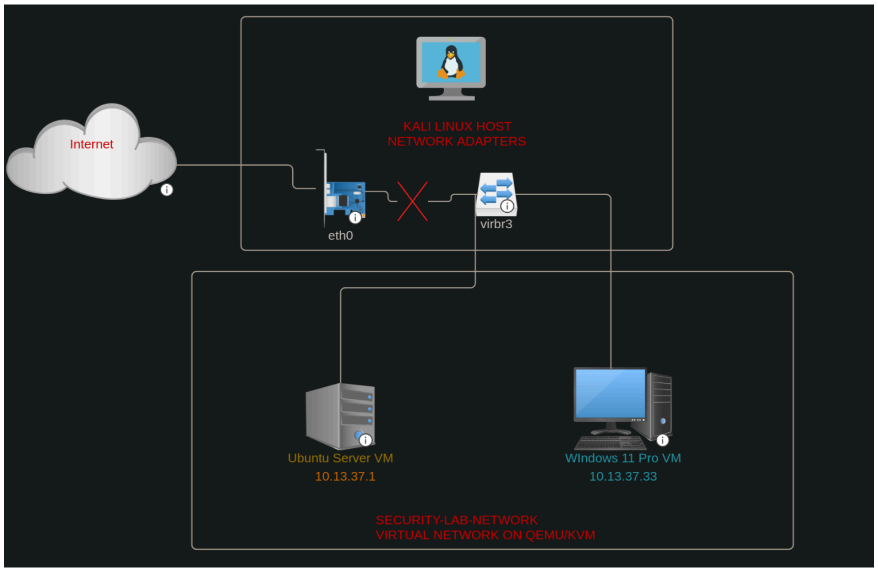


ARP vs. Ping:

Ping uses ICMP (Internet Control Message Protocol), a network layer protocol, while ARP is a data link layer protocol used to find the MAC address of a device with a given IP address.

Robert Carpenter
github.com/robertmcarpenter
April 4th 2025

My Virtual Network now looks like this:



Each Network Environment in the topology diagram above is denoted by the machines / adapters inside their own rectangle. Note that even though virbr3 is a virtual network adapter on my Host machine, it functions as a layer 2 switch for the VMs connected to it.

In the Next lab, I will configure the Network so that my Ubuntu VM acts as the Layer 3 switch and router for the virtual network!

This will now conclude this lab.