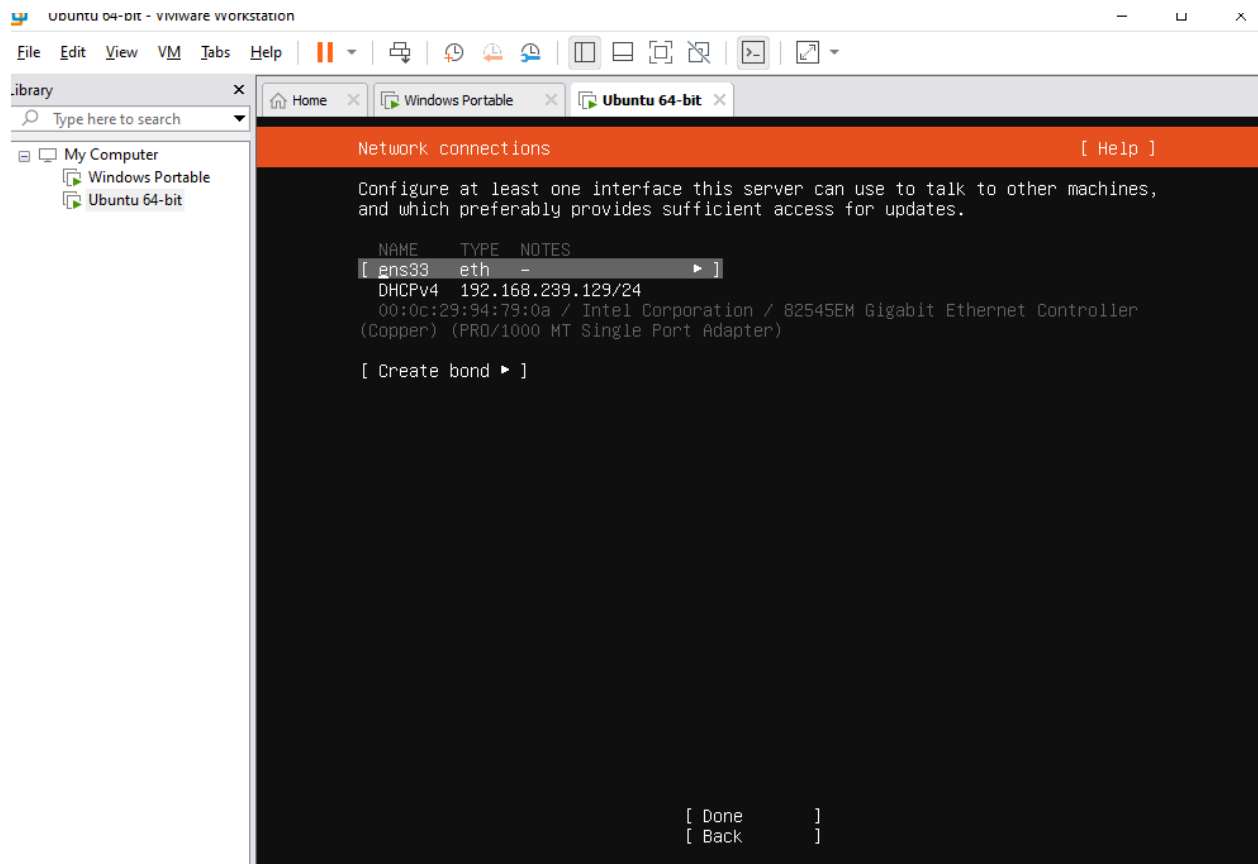


Network: vmnet8
Subnet IP: 192.168.239.0
Subnet mask: 255.255.255.0
Gateway IP:

Set up your virtual environment

Download and install Ubuntu into a new VM

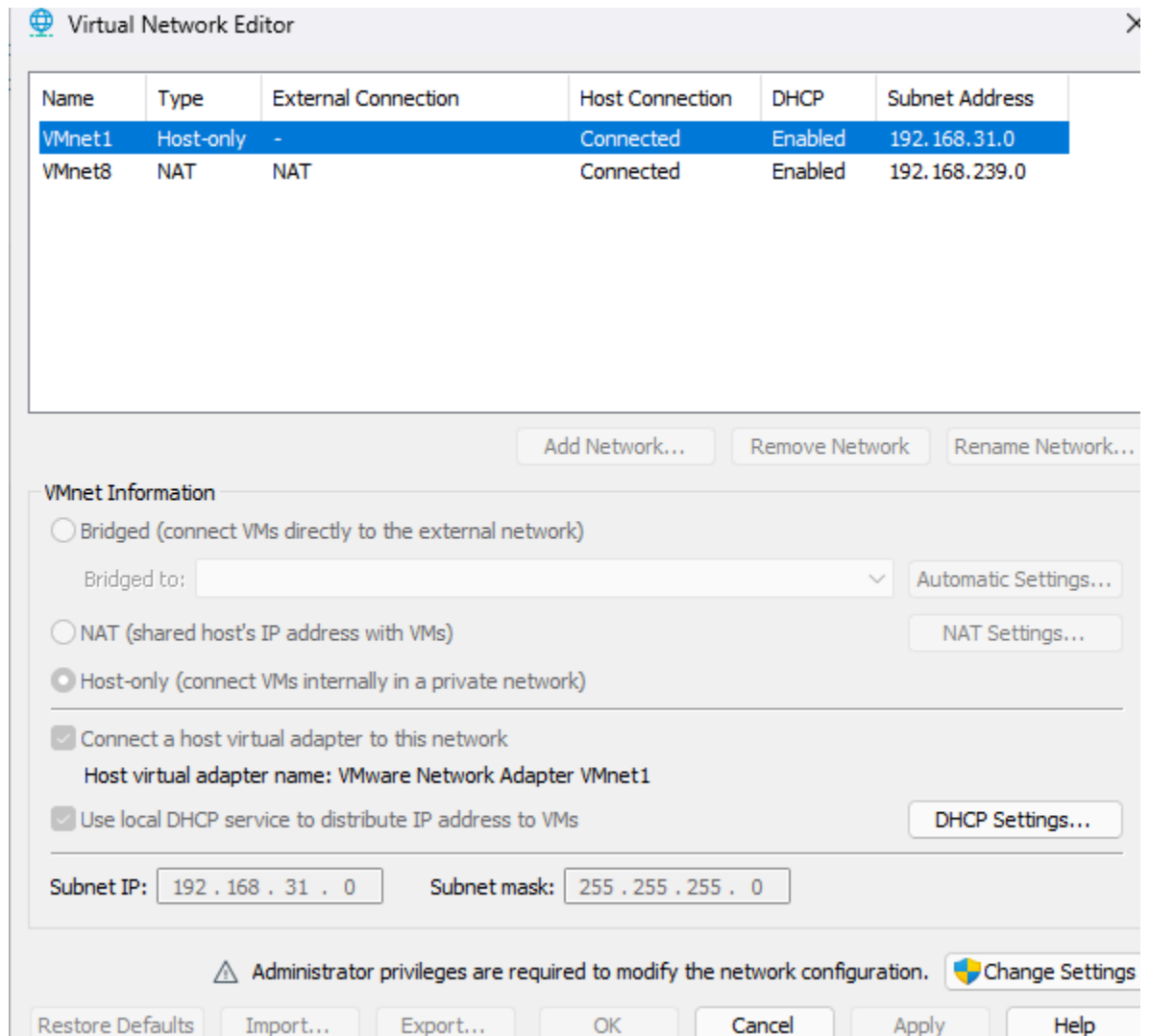
- a. Download the [Ubuntu Server 22.04.1](#) installer ISO.
 - i. NOTE: I am specifying the **SERVER** version of Ubuntu because it comes preinstalled with necessary packages. If you choose the Desktop flavor, you will have issues, and you are wasting unnecessary resources.
- b. Once downloaded, [create a new VM in Workstation](#) with the following specs
 - i. Use the downloaded ISO as the installer image
 - ii. 14GB Disk size
 - iii. Customize Hardware
 - 2 CPU cores
 - 2GB RAM
 - iv. During OS install, leave defaults unless otherwise specified
 - Use Tab to navigate, Space to check boxes, Enter to confirm
 - “Installer update available”
 - a. “Continue without updating”



When you get to “Network connections” section above, we need to take a few steps to set a static IP address for this VM so that it doesn’t change throughout the lab or beyond it.

- a. Find out the gateway IP of your VMware Workstation NAT network
 - i. In VMware Workstation, click “Edit” menu at top
 - ii. Click “Virtual Network Editor”
 - iii. Select the “Type: NAT” network

iv. Click “NAT Settings...”



The screenshot shows the 'Virtual Network Editor' window. At the top, there is a table listing two virtual networks: VMnet1 and VMnet8. VMnet1 is a 'Host-only' network connected to the host, with a subnet of 192.168.31.0. VMnet8 is a 'NAT' network connected to the host, with a subnet of 192.168.239.0. Below the table, there are buttons for 'Add Network...', 'Remove Network', and 'Rename Network...'. The 'VMnet Information' section for VMnet1 shows three radio button options: 'Bridged' (selected), 'NAT', and 'Host-only'. The 'Bridged' option is selected, and the 'Bridged to:' dropdown is set to 'Automatic Settings...'. The 'NAT' option is disabled, and the 'Host-only' option is also disabled. Below these options, there are two checked checkboxes: 'Connect a host virtual adapter to this network' (with the adapter name 'VMware Network Adapter VMnet1') and 'Use local DHCP service to distribute IP address to VMs'. There are buttons for 'Automatic Settings...', 'NAT Settings...', and 'DHCP Settings...'. At the bottom, there are input fields for 'Subnet IP' (192.168.31.0) and 'Subnet mask' (255.255.255.0). A warning message at the bottom states 'Administrator privileges are required to modify the network configuration.' with a 'Change Settings' button. At the very bottom, there are buttons for 'Restore Defaults', 'Import...', 'Export...', 'OK', 'Cancel', 'Apply', and 'Help'.

Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet1	Host-only	-	Connected	Enabled	192.168.31.0
VMnet8	NAT	NAT	Connected	Enabled	192.168.239.0

Buttons: Add Network..., Remove Network, Rename Network...

VMnet Information

☐ Bridged (connect VMs directly to the external network)
Bridged to: [Automatic Settings...] Automatic Settings...

☐ NAT (shared host's IP address with VMs) NAT Settings...

☐ Host-only (connect VMs internally in a private network)

☒ Connect a host virtual adapter to this network
Host virtual adapter name: VMware Network Adapter VMnet1

☒ Use local DHCP service to distribute IP address to VMs DHCP Settings...

Subnet IP: 192.168.31.0 Subnet mask: 255.255.255.0

⚠ Administrator privileges are required to modify the network configuration. Change Settings

Buttons: Restore Defaults, Import..., Export..., OK, Cancel, Apply, Help

- v. Copy down the “Subnet IP” & “Gateway IP”, we’ll need it in the next step

NAT Settings

Network: vmnet8

Subnet IP: 192.168.239.0

Subnet mask: 255.255.255.0

Gateway IP: 192.168.239.2

- vi.
- vii. Close the NAT Settings and Virtual Network Editor windows.

- b. Now, back in the Ubuntu installer, let’s change the interface from DHCPv4 to Manual.

Network connections [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
[ens33]	eth	-
DHCPv4 192.168.239.129/24		
00:0c:29:94:79:0a / Intel Corporation Ethernet Controller (Copper) (PRO/1000 MT Single Port)		
[Create bond ►]		

- ◀ (close)
- Info
- Edit IPv4
- Edit IPv6
- Add a VLAN tag

i.

Edit ens33 IPv4 configuration

IPv4 Method:

- Automatic (DHCP)
- Manual
- Disabled

[Cancel]

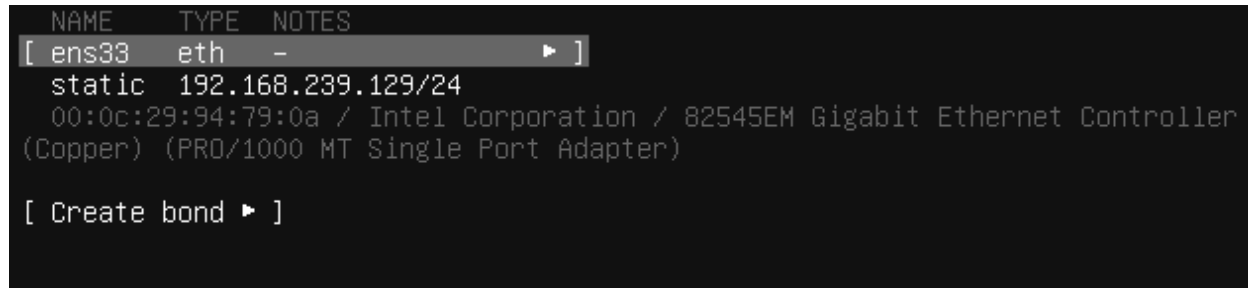
ii.

- iii. Be sure to carry forward the subnet and gateway IP from the previous step V., but adding the /24 to subnet IP. The

“Address” gets copied from what was previously assigned via DHCP.

iv. 

v. When you’re done, you should see this.

vi. 

- i. **NOTE:** Write down the Linux VM’s IP address because you will need it multiple times throughout this

c. Set a memorable username/password (this is just a lab)

- i. Your name: `user`
- ii. Your server's name: `attack`
- iii. Username: `user`
- iv. Password: `password`

Profile setup [Help]

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

Your name:

Your server's name:
The name it uses when it talks to other computers.

Pick a username:

Choose a password:

Confirm your password:

v.

d. Install OpenSSH server?

- i. `[check/yes]`

SSH Setup

You can choose to install the OpenSSH server package to enable secure remote access to your server.

☒ Install OpenSSH server

Import SSH identity: [No ▼]
You can import your SSH keys from GitHub or Launchpad.

Import Username:

☒ Allow password authentication over SSH

ii.

e. Continue installing OS until “Install complete!”

f. Hit Enter on `[Reboot Now]`

- i. If it hangs on “removing the CDRom” just press Enter

g. After the reboot, let's perform a quick connectivity check.

- i. Logon with the credentials we defined during install
 - i. Username: `user`
 - ii. Password: `password`
- ii. Make sure DNS and outbound pings are working
`ping -c 2 google.com`
- iii. If your output resembles mine, you're good to go.

```
user@attack:~$ ping -c google.com
ping: invalid argument: 'google.com'
user@attack:~$ ping -c 2 google.com
PING google.com (142.251.10.102) 56(84) bytes of data.
64 bytes from sd-in-f102.1e100.net (142.251.10.102): icmp_seq=1 ttl=128 time=9.90 ms
64 bytes from sd-in-f102.1e100.net (142.251.10.102): icmp_seq=2 ttl=128 time=10.2 ms

--- google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1007ms
rtt min/avg/max/mdev = 9.898/10.051/10.204/0.153 ms
user@attack:~$ _
```

Setup Attack System

We'll perform these steps from your host system, by using SSH to access the Linux VM.

Use an SSH client to access the Ubuntu VM so that you can easily copy/paste commands. MacOS/Linux/Modern Windows systems have built in SSH abilities, but there are [third party tools](#) for this as well.

1. Using the statically assigned IP address we copied down in the Linux VM installation process, let's SSH onto the VM from your host system to make future CLI activities easier thanks to copy/paste magic.
 - a. I'll let you decide which SSH client to use, but from a modern Mac/Linux/Windows system, simply open a command prompt and run
`ssh user@192.168.239.129` (Use your linux IP address, this is mine.)
1. Now, from within this new SSH session, proceed with the following instructions to setup our attacker C2 server. First, let's drop into a root shell to make life easier.
`sudo su`

Run the following commands to download [Sliver](#), a Command & Control (C2) framework by BishopFox

```
# Download Sliver Linux server binary
```

```
Wget
```

```
https://github.com/BishopFox/sliver/releases/download/v1.5.34/sliver-server_linux -O /usr/local/bin/sliver-server
```

```
# Make it executable
```

```
chmod +x /usr/local/bin/sliver-server
```

```
# install mingw-w64 for additional capabilities
```

```
2. apt install -y mingw-w64
```

Now let's create a working directory we'll use in future steps

```
# Create our future working directory
```

```
3. mkdir -p /opt/sliver
```