Fundamental Skills - Setting up a Virtual Machine

Category	Experience Level
Misc	Novice

Contents

Intro

Virtual Machines (VMs) are useful tools for running a 'virtualised' version of a computer as a program on your normal computer.

What does this mean?

- you can run a machine with a different Operating System (OS) to your normal computer; for example, you could run [Kali Linux](#)-(a-common-os-that-comes-with-lots-of-cybersecurity-tools) as a sub-program on your normal Windows Computer, without having to worry about Dual Booting or Live USBs
- you can create machines for dangerous activities like exploit development and malware analysis, or just to experiment with a new OS, and be able to safely discard them or revert them to clean states
- you can use machines prebuilt and configured by experts for specific purposes, such as Tracelabs' [Open-Source Intelligence VM](#)

- you can create internal networks of machines to practice hacking interconnected machines, such as an Active Directory forest or a mailserver
- you can adjust machine specifications, such as amounts of RAM, on the fly (within the limits of your actual computer!)

Virtualisation Software

You'll need a piece of software that is capable of running Virtual Machines. Some Operating Systems, such as Windows 10 Pro, have such software preinstalled. Most people, however, will need to install the software separately.

We will be teaching VirtualBox in this lesson. You can download it from here:

https://www.virtualbox.org/wiki/Downloads

You will need to pick the right version for your **host** Operating System (the OS of your actual computer). You can then start creating Virtual Machines with many different **guest** Operating Systems.

You'll also need to **enable virtualisation** on your computer - this can often be done in the BIOS, which is an interface that controls many internal aspects of your computer. It can usually be accessed during boot time, but the method for accessing it will vary based on your computer model. Searching "enable virtualisation on *your computer here*" into Google should do the trick!

A good alternative to VirtualBox is VMWare - it is feature rich and has a very clean User Interface, but is a paid product. If you think you'll only need to run one Virtual Machine, you can download the free VMWare Player software - but beware, we may ask you to install other Virtual Machines in our sessions!

Basic VirtualBox Usage

The main screen of VirtualBox lists all your VMs - clicking a VM shows more details:
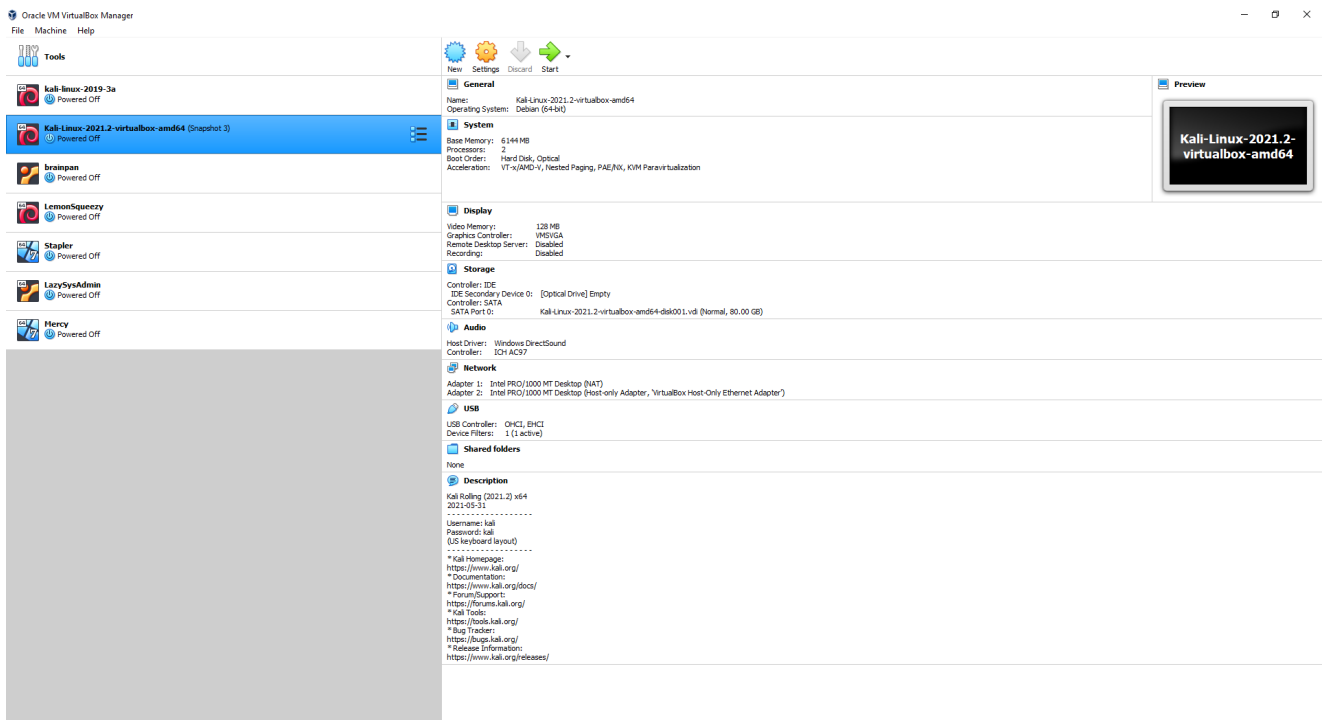
To power on a VM, simply click the green *Start* button. This will start the machine in a new window - you can see that my *host* OS is Windows, but the VM is booting in GRUB (the boot loader for the *guest* OS, Debian):
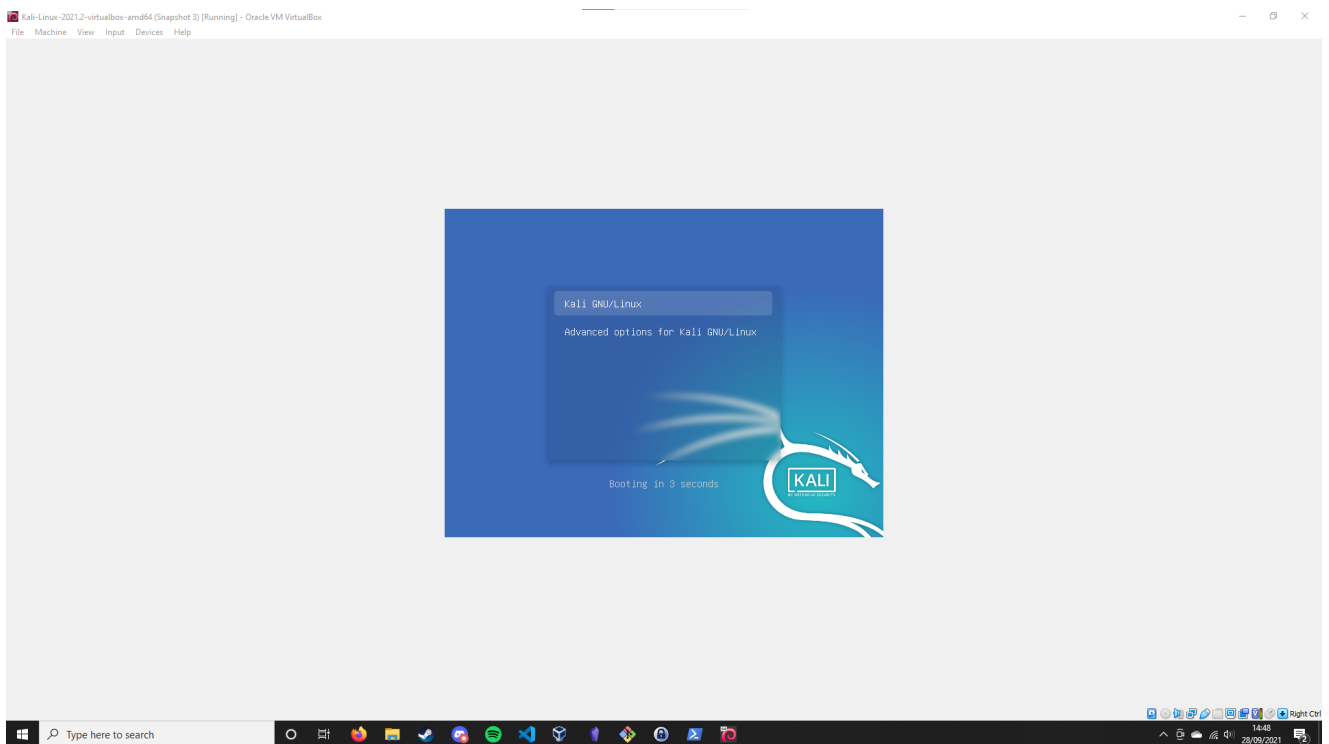
Powering off a VM is as simple as clicking *File > Close*, or powering it off as normal using the power settings in the machine itself:
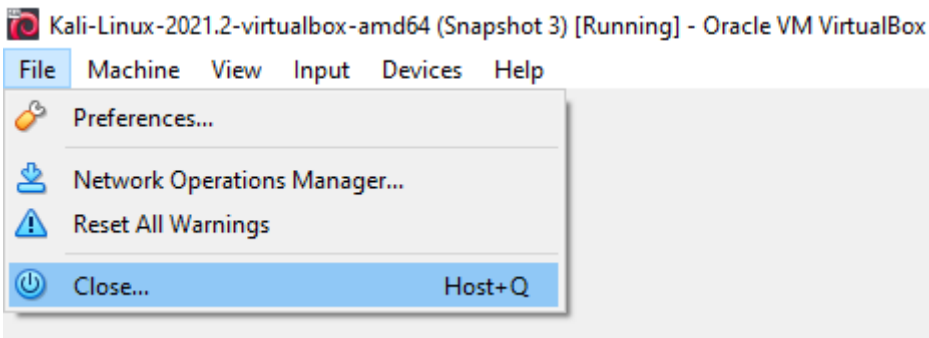
Virtual Box will *capture* your mouse and keyboard by default, which means it will direct your mouse and keyboard to the VM when the window is active. To *release* your controls simply press the *Host Key* (`Right Ctrl` by default on Windows) - this will let you interact with the Host OS again.

To make your VM full screen, simply press `Host + F` (where `Host` is your *Host Key*).

You can also pause the machine (`Host + P`), take snapshots (`Host + S`), and revert to an earlier state.

Snapshots are a powerful feature that let you save the complete machine state and go back to it later - for example, before running `sudo apt-get dist upgrade` so you can return to your old distribution if something goes wrong.

You can view snapshots by clicking a machine's hamburger menu and selecting the *Snapshots* tab:
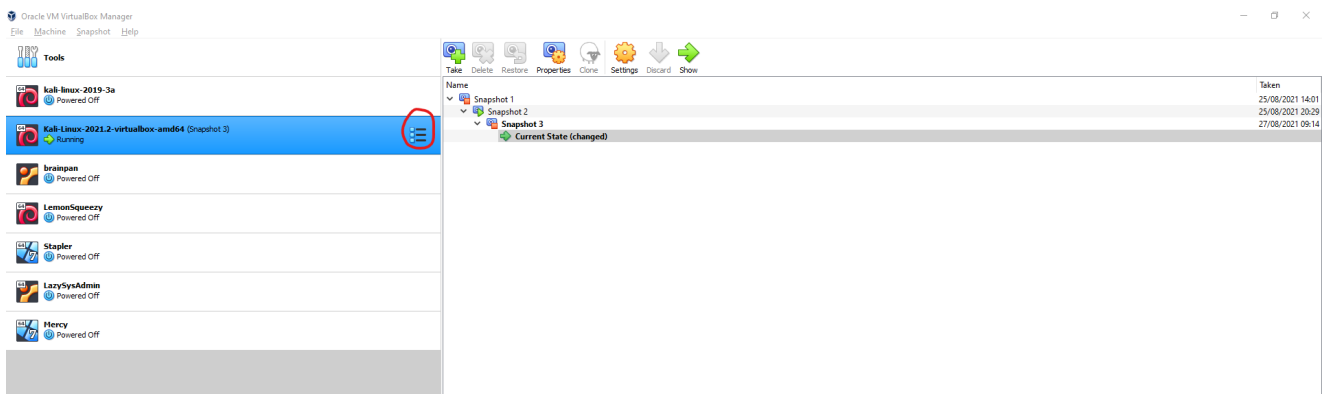
From there you can restore to a specific snapshot, delete a snapshot, and more.

A fresh install won't have any VMs listed - we'll discuss how to add them in the next section, but we first need to go over different VM formats.

Installing a VM

We've shown you what VirtualBox looks like, but not how to actually *install* a Virtual Machine.

What Should I Choose?

There are a few options when setting up a virtual machine.

Perhaps the easiest option is to use a **machine image**, which is a fully configured Virtual Machine that can be loaded straight into VirtualBox and ran straight away. However, machine image files can be very large and slow to download. Choose this option if you want your machine completely configured for you.

The other option is to use a **disk image** to install the guest operating system from. This is a little like installing a fresh copy of Windows from an install disk. It gives you more control over the installation process, but is slower to install once downloaded. Choose this option if you're comfortable following an install wizard and want extra control over your installation.

Downloading a Machine Image

We'll use Kali Linux as an example - Kali is a powerful Operating System built for penetration testing, and comes preconfigured with a large suite of tools. Don't be too overwhelmed by everything that's on it - get used to using a terminal first!

You can get a Kali Machine image from here: https://www.kali.org/get-kali/#kali-virtual-machines. Select the version for VirtualBox, and wait for the download to complete (it may take a while):
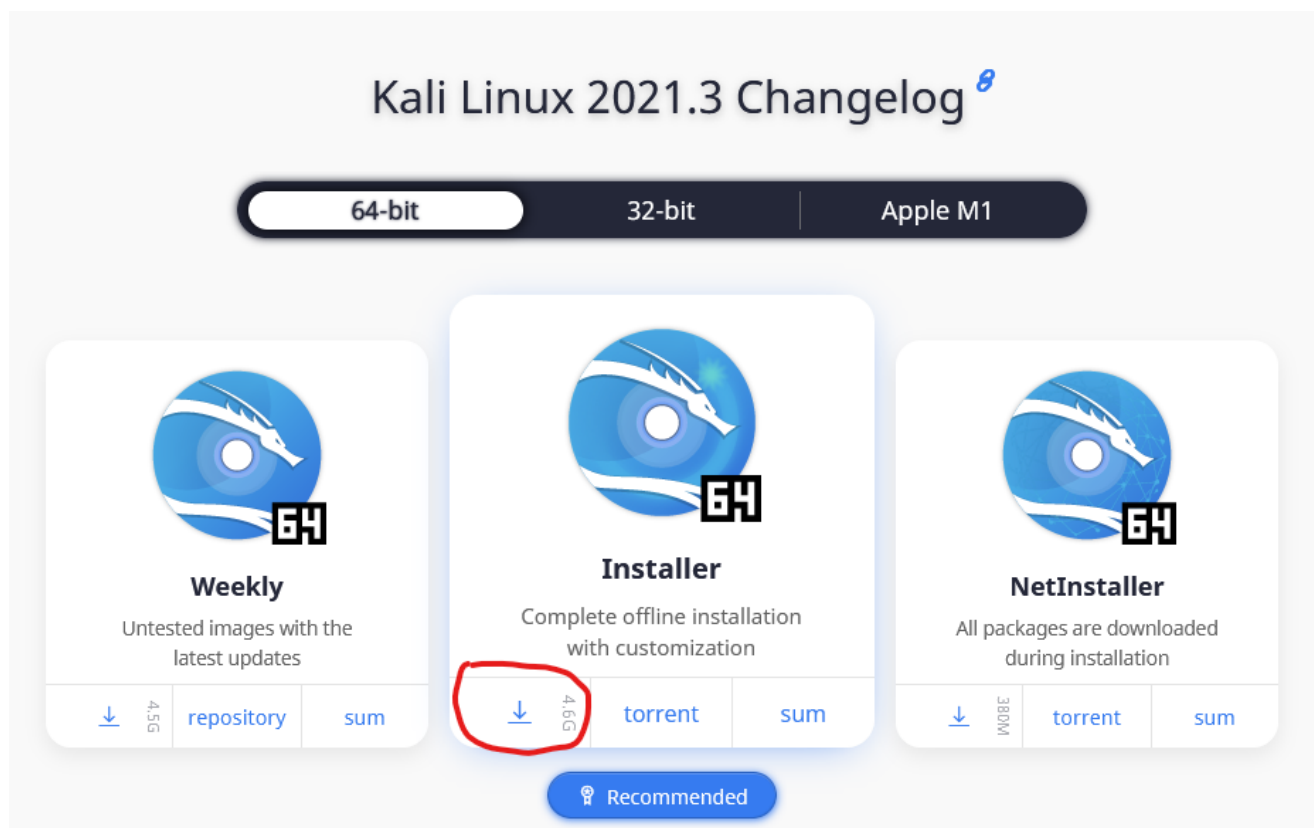

This will download as a large `.ova` file.

Downloading a Disk Image

If you prefer a Disk Image, you can get the Kali Disk Image from here: https://www.kali.org/get-kali/#kali-bare-metal:
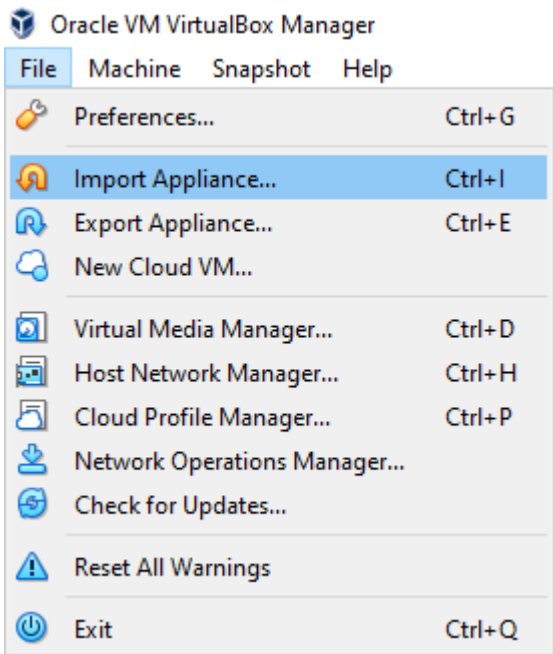

This will download as a `.iso` file.

Installing with Virtual Box

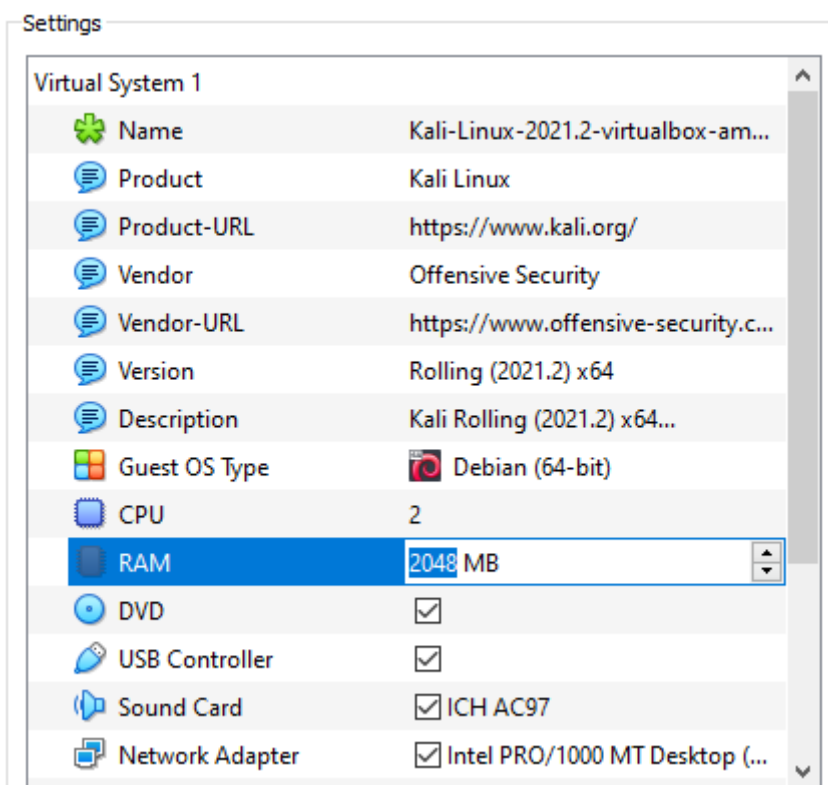Once the downloads are complete, you can import whatever VM format you downloaded into VirtualBox.

Installing from a Machine Image

To import a Disk Image, press `Ctrl + I` or go to *File > Import Appliance*:
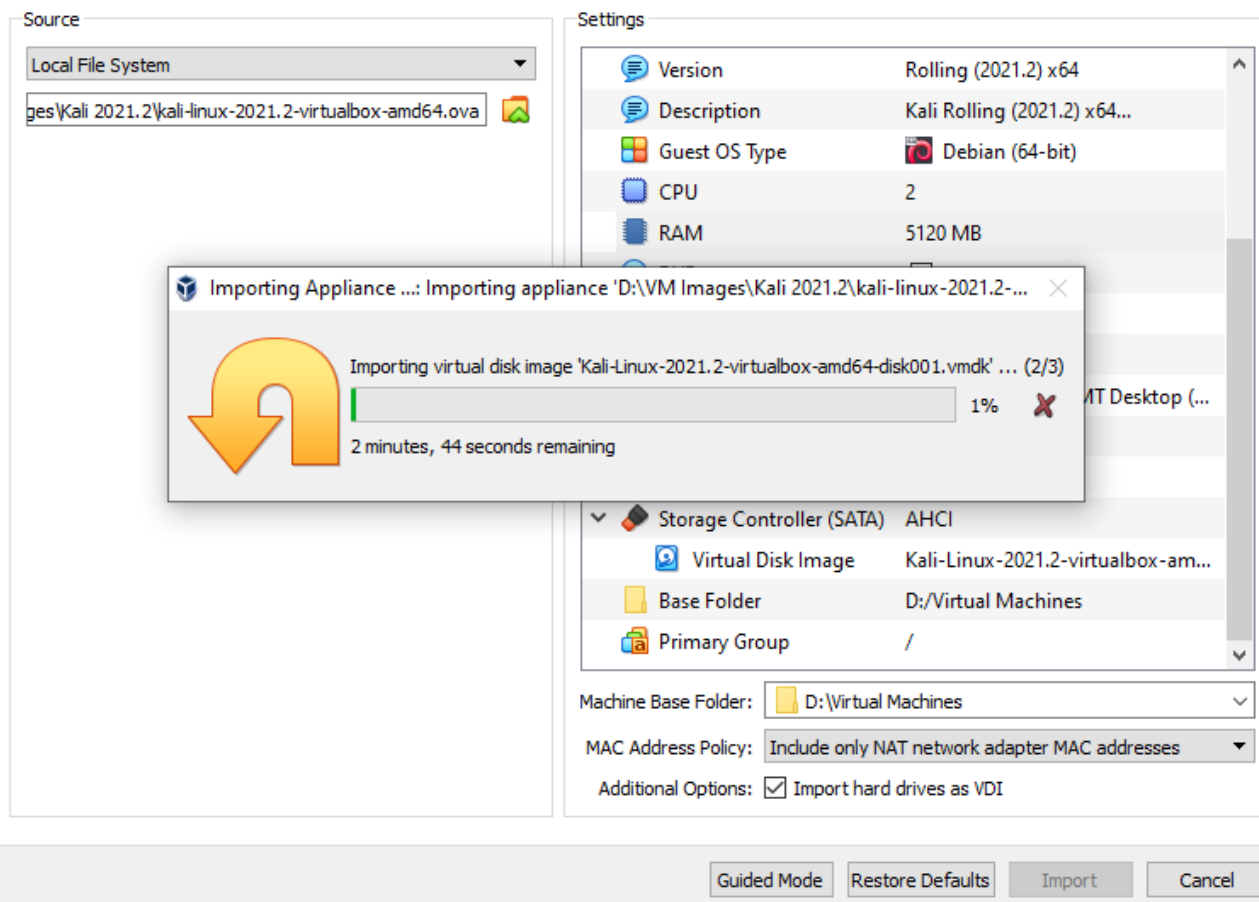
Then select the downloaded Machine Image - it should have the `.ova` extension.

You can edit the settings, such as to give it more RAM:

You can also choose where to install the VM to - if you have a smaller SSD drive and a larger HDD, it may be a good idea to install it to the HDD.

Then press *Import*, and VirtualBox will load your machine!

Most Kali images have around 80GB of disk space assigned, which should be more than enough for most people - this space won't be consumed on your drive immediately, it is simply an upper limit of the size of the VM.

> Security Tip: we also recommend changing the default password on your kali image - by default, the credentials are kali:kali
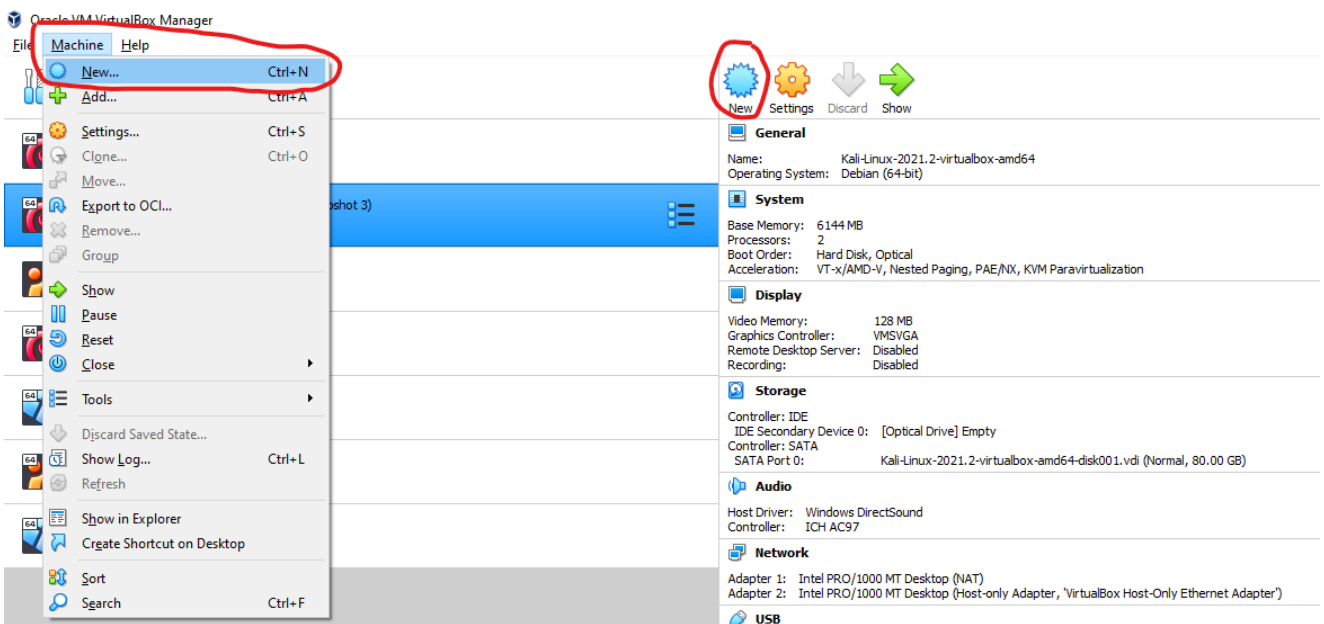
Other Formats

Some Machine Images may be in a format built for VMWare, especially if they were created in VMWare originally. For example, here is the machine Lemonsqueezy from Vulnhub:
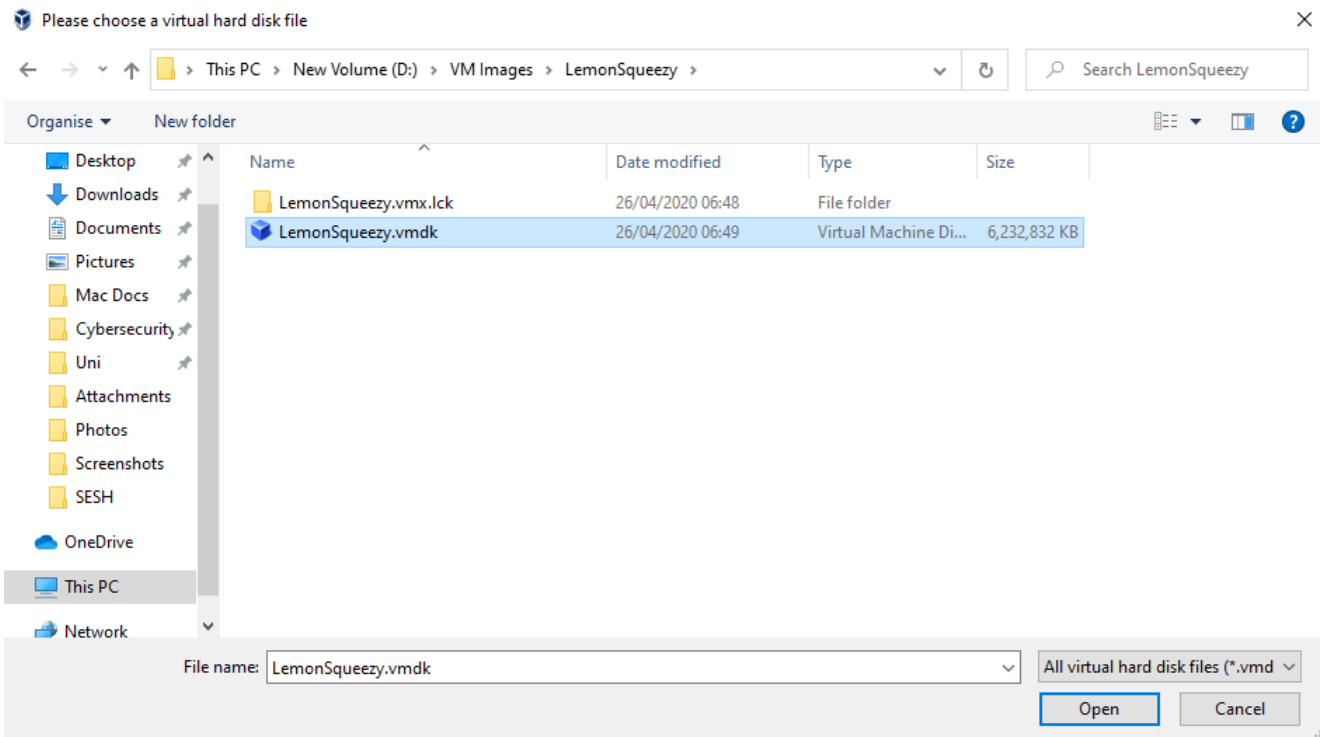
VirtualBox can still load these files, but the process is slightly different. First, press `Ctrl + N` or the *New* button to create a new machine:

Then give your machine a name and install location, and assign it some RAM:
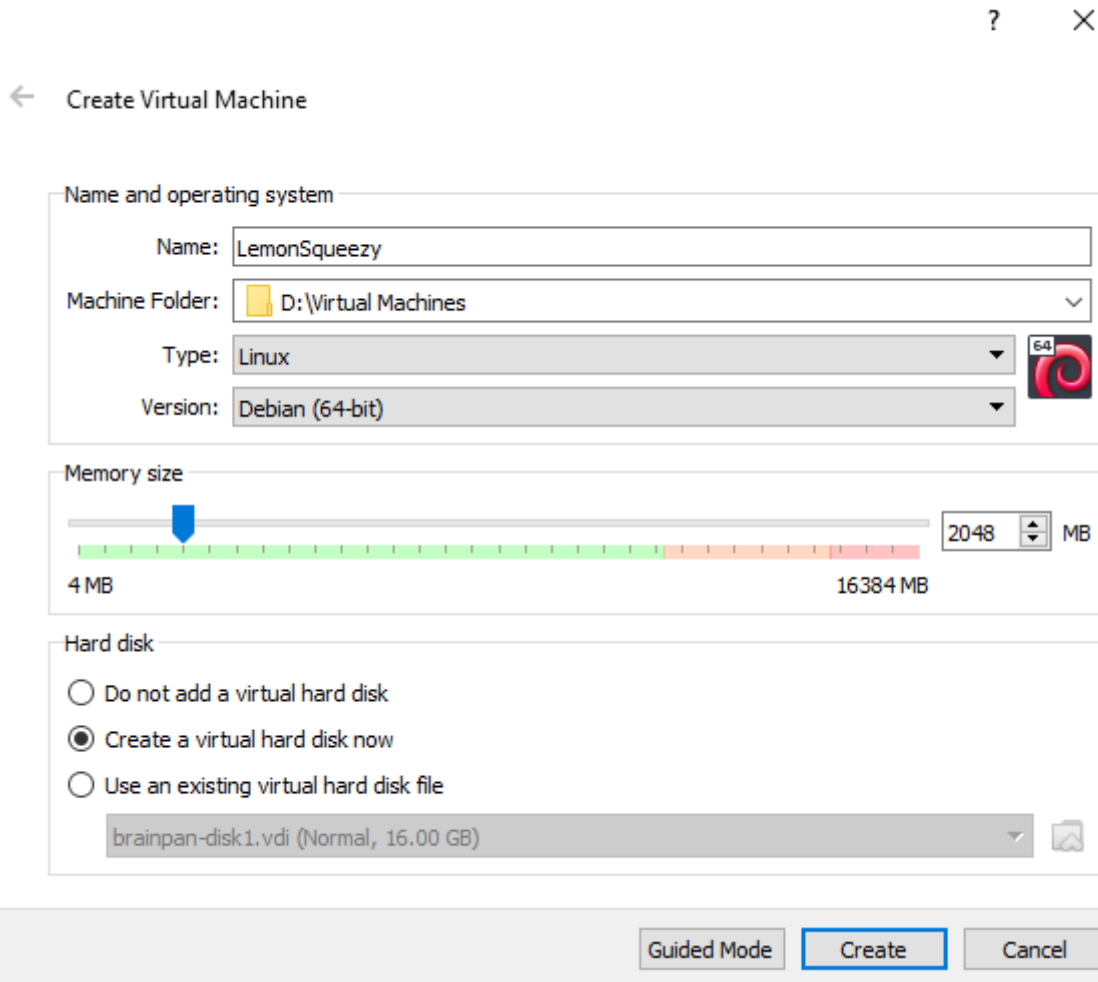
In the bottom, select *Use an existing virtual hard disk file* and click the folder icon to select the `.vmdk` file for the machine you're importing - if it's not in the list, click *Add* to import it:
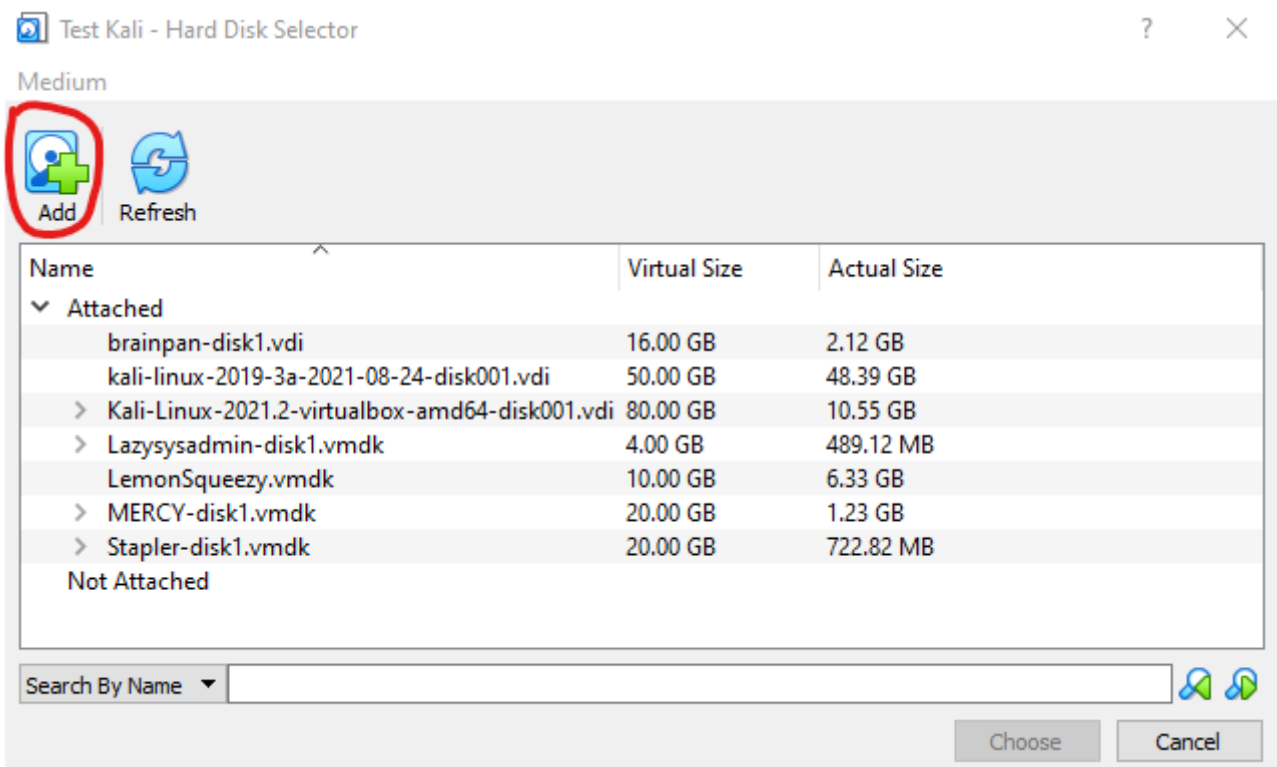
The image should now be selected:

Then click create, and your machine should be ready to go.

Installing from a Disk Image

Similarly to installing from a `.vmdk` file in the above example, installing from a Disk Image requires creating a new machine (rather than importing one).

To import a downloaded Kali disk image (`.iso` file), hit `Ctrl + N` to create a new machine, select the Linux Operating system, and give it whatever specifications you want, but select the *Create a virtual hard disk now* option:

Decide what size virtual hard disk you want (we recommend at least 25GB):

Now create the machine and go to Settings - from here, change the machine's boot order under the *System* tab:
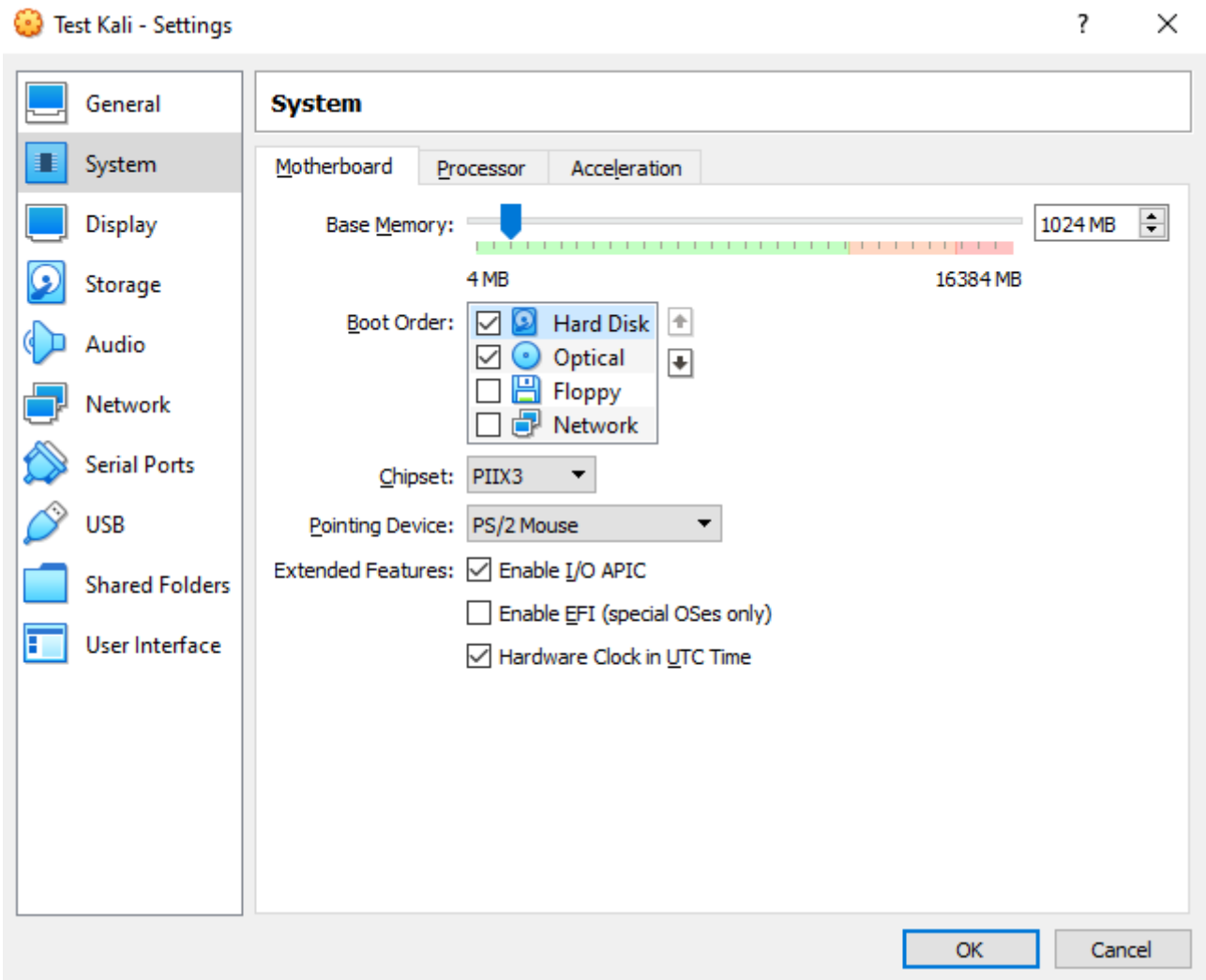
Then you must import the disk image you downloaded into the VM's optical drive - to do this, go to the *Storage* tab and click the entry that says *Empty*:
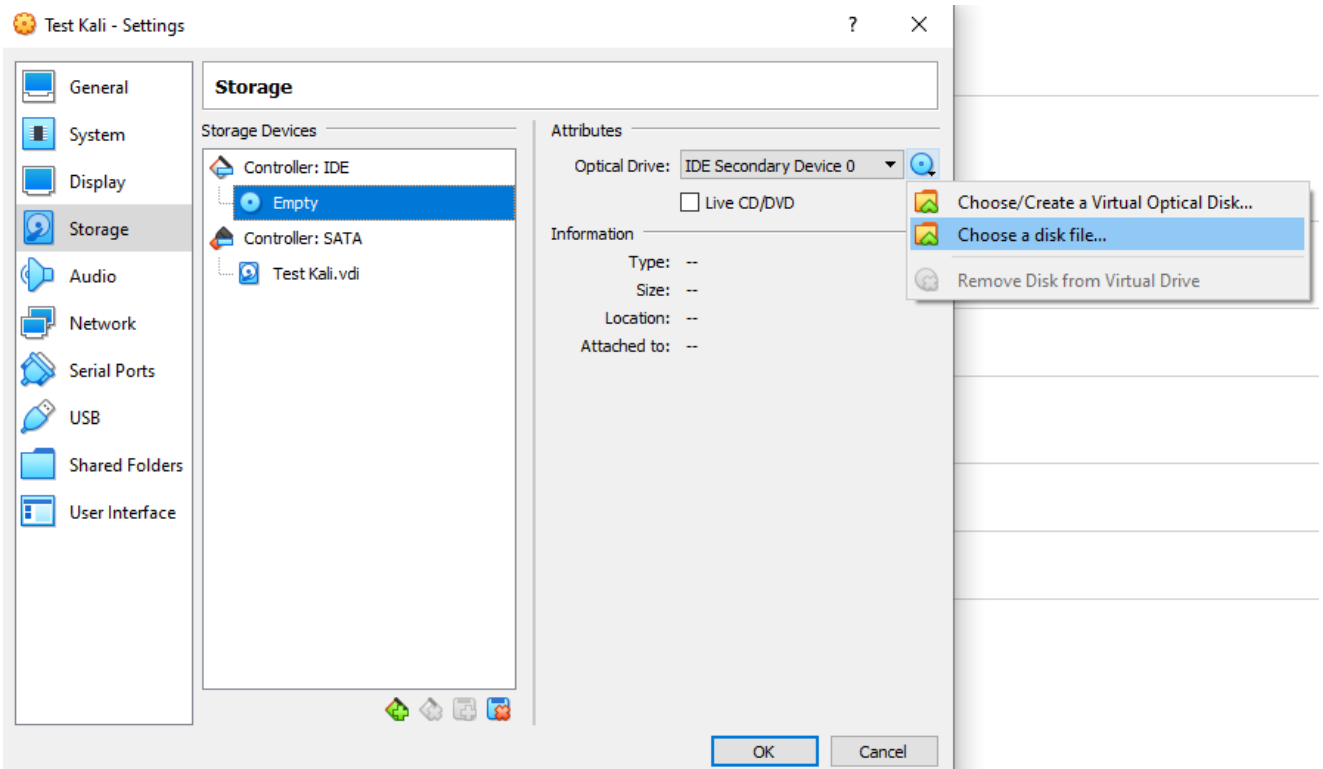
Select your `.iso` file from the file explorer, and then press *Start* on the machine. It should now boot from the disk image you just selected, and start the installation process!

Follow the install wizard, selecting the keyboard layout and timezone settings you want, creating a new user, and configuring your disks - Kali's [official documentation](#) has a great guide to doing this, and we won't look at it here in the interests of keeping the guide short. If you don't want to go through the installation process, you should use a Machine Image instead.

Configuring the Machine

You can change the amount of RAM on your machine at any time by visiting the machine settings. There are a number of other options available, such as using a bidirectional clipboard.

For some features to work, including two-way copy and paste, you will need the VirtualBox Extension Pack - you can download it here:
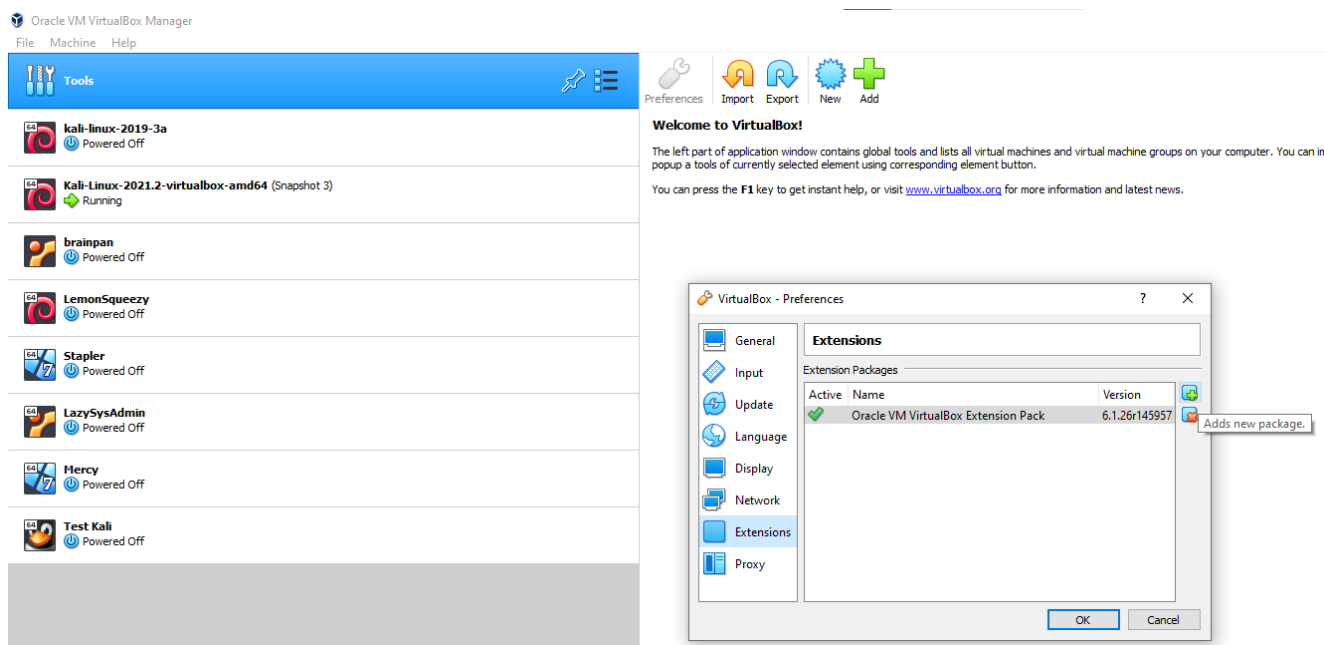https://www.virtualbox.org/wiki/Downloads:

After you've downloaded it, import them by going to *Tools > Preferences > Extensions* and clicking the `+` button:


Networking a VM

This section is not necessary for a basic VM setup, but will be useful if you intend to connect two VMs together. You might, for example, want to have a Kali VM and a

vulnerable Virtual Machine from VulnHub to attack.

Having the right networking settings is important, as you don't want to have a vulnerable virtual machine exposed to the internet!

To connect two VMs together, the easiest option is to use a VirtualBox Host-Only adapter.

Your Kali VM, for example, will want one adapter still connected to the internet - use a NAT adapter for this:
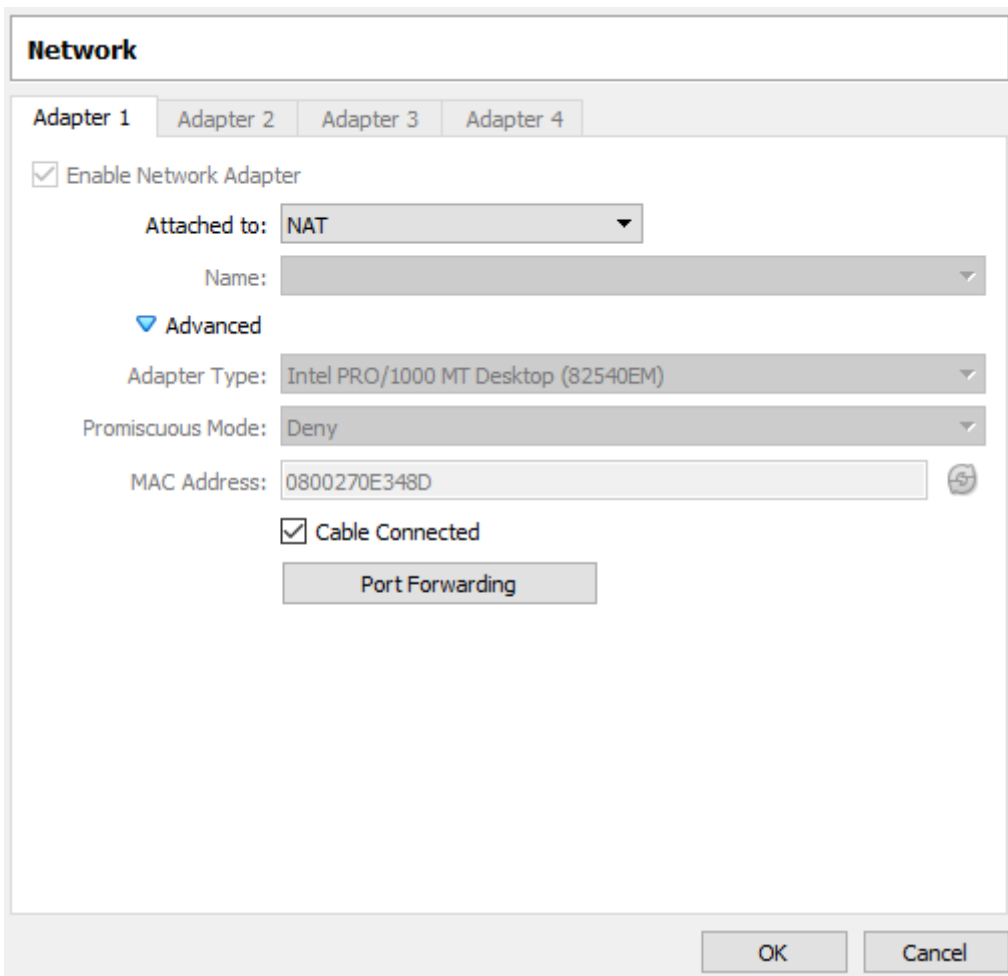
The second adapter for the Kali machine should be a host-only adapter:
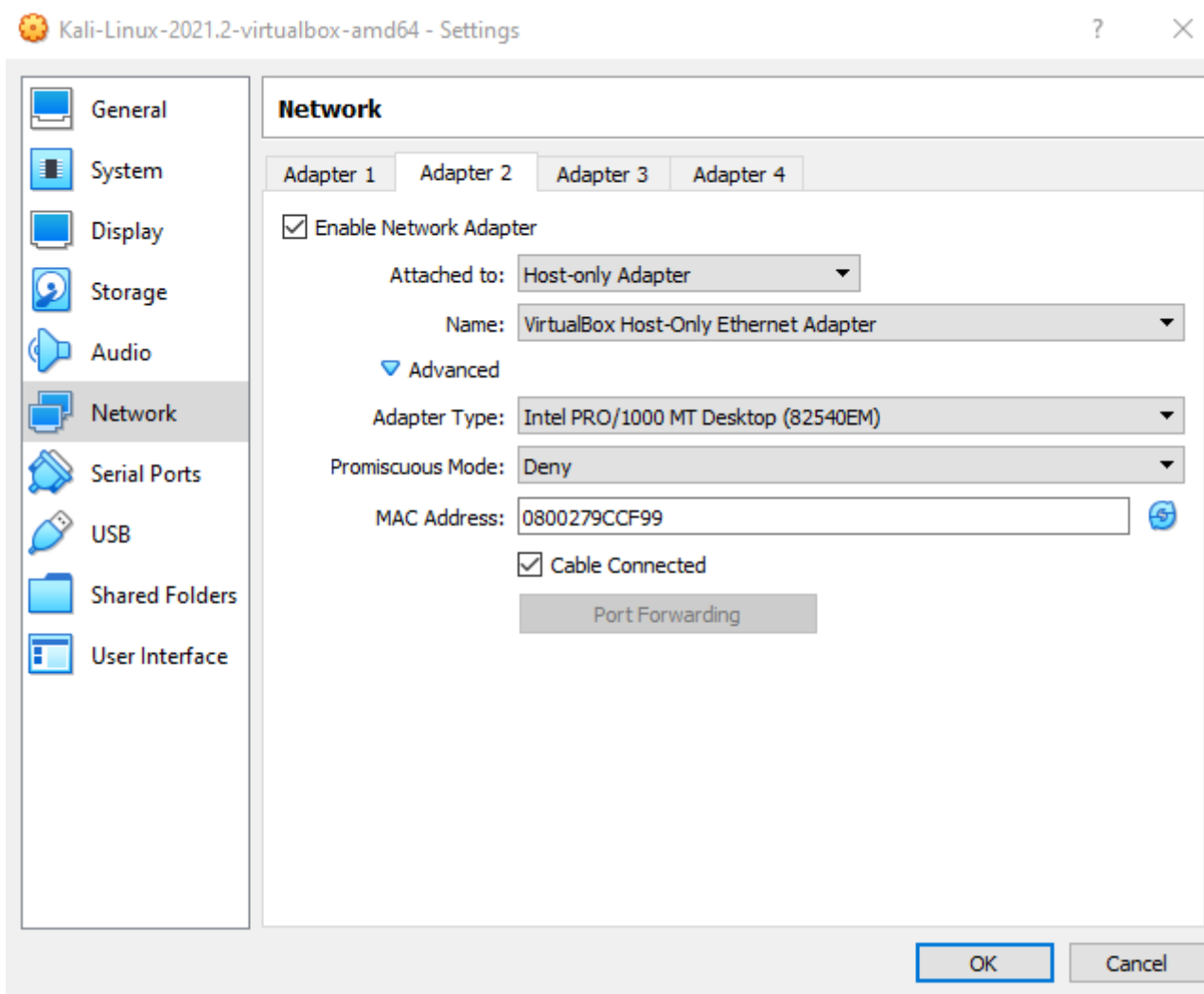
The vulnerable machine should have **only** a host-only adapter, so that it can only communicate with other VMs (and not the internet):

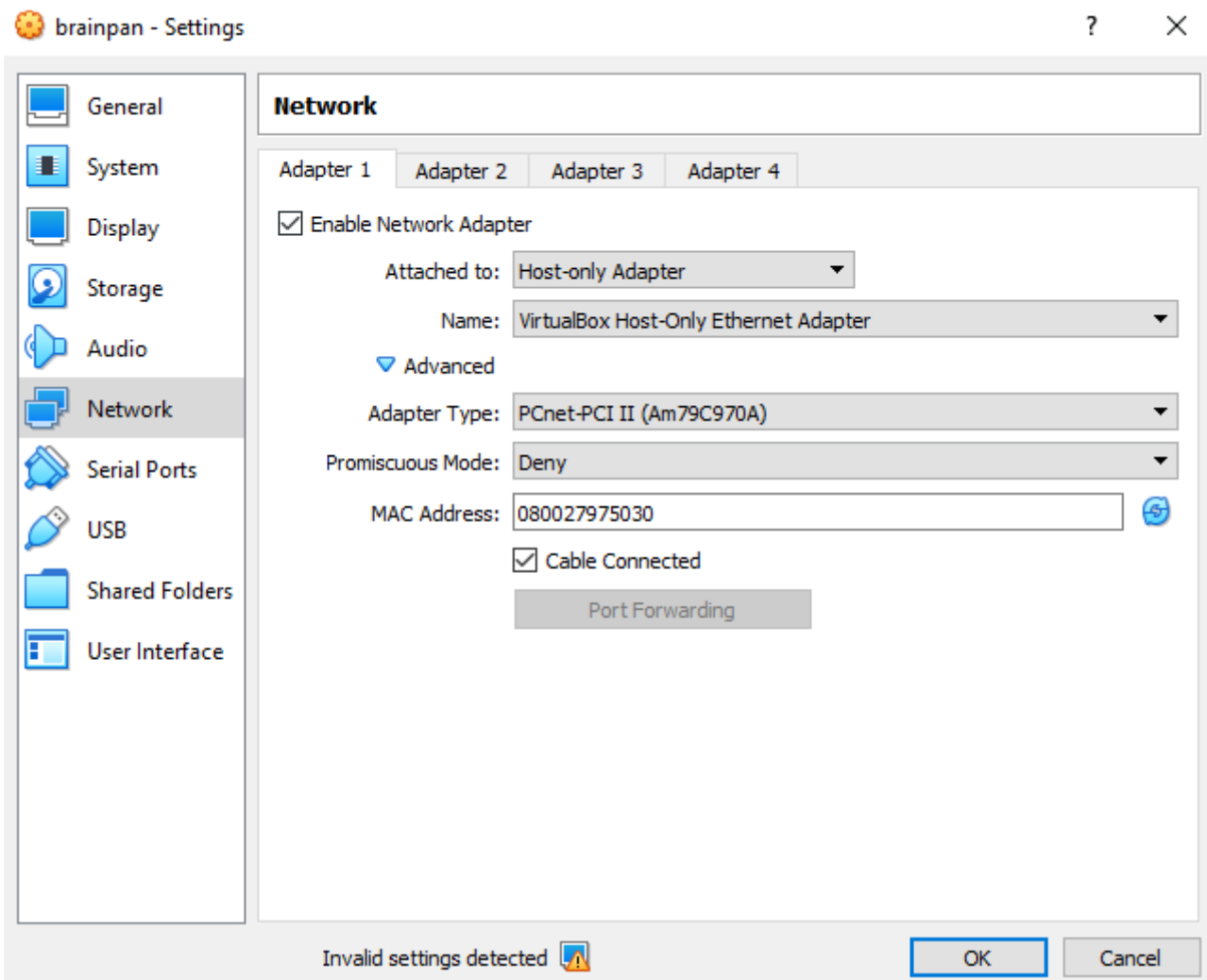The final step will be to make the host-only interface on Kali automatically assign IP addresses - to do this, edit your `/etc/network/interfaces` file:

```
$ sudo nano /etc/network/interfaces
```

The edited file should look like this, with the `eth1` interface assigning IPs automatically:

```
# The loopback network interface
auto lo
iface lo inet loopback


# Dedicated eth1 for internal/NAT lab interface
auto eth1
iface eth1 inet dhcp
```

You should now be able to scan the `eth1` interface and discover the IP of your vulnerable machine!

```
┌──(kali㉿kali)-[~]
└─$ sudo arp-scan --interface eth1 192.168.56.102/24
Interface: eth1, type: EN10MB, MAC: 08:00:27:9c:cf:99, IPv4: 192.168.56.102
WARNING: host part of 192.168.56.102/24 is non-zero
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-
scan)
192.168.56.1     0a:00:27:00:00:12      (Unknown: locally administered)
192.168.56.101   08:00:27:97:50:30      PCS Systemtechnik GmbH
192.168.56.100   08:00:27:9e:b0:fe      PCS Systemtechnik GmbH


3 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 2.025 seconds (126.42
hosts/sec). 3 responded
```

> Above: an arp-scan showing the vulnerable machine with IP address
> 192.168.56.101

An alternative command to scan this interface is `nmap -sn 192.168.56.0/24`.

Learn more about VirtualBox networking types here:
https://www.virtualbox.org/manual/ch06.html

Worksheet

1. Install a Kali Virtual Machine, either using a Machine Image or a Disk Image
2. Install Pinky's Palace from VulnHub: https://www.vulnhub.com/entry/pinkys-palace-v1,225/ - this is an OVA file, so it can be imported as a Machine Image
3. Network Kali and Pinky's Palace together with host-only adapters, and discover the IP for Pinky's Palace