



HACKINGDOJO.COM LAB MANUAL

Mukyu (1R)

ABSTRACT

Lab companion for the HackingDojo.com Mukyu (1R) course. Intended to be used as part of the student membership to the Hacking Dojo.

WILHELM, THOMAS

Table of Contents

1	Introduction	2
2	The Virtual Machine	3
2.1	Installation.....	3
2.2	Installing Kali	12
3	Pentest Target.....	29
3.1	De-ICE 1.100	29

1 INTRODUCTION

Welcome to the Hacking Dojo Mukyu (1R) Personal Pentest Lab manual. This document is not intended to be a way to learn professional penetration testing as a standalone tool. The examples provided in this document are intended to guide students towards creating their first virtual lab, to be used in conjunction with the video and online training provided at HackingDojo.com. However, if you are viewing this and are not a member of the Hacking Dojo, welcome!... it means someone who was a member thought you might enjoy this document, and the staff at HackingDojo.com hope you do as well.

The purpose of this document is to walk a student through setting up their own virtual lab, and to be able to complete the homework provided in the student training. If you have any issues with this document, please don't hesitate to reach out to our support team on Skype to help. Enjoy!

2 THE VIRTUAL MACHINE

This document was written specifically for students beginning their journey into the field of professional penetration testing. The instructions here are intended to begin that journey, and are written for students who have no experience with Linux/Unix operating systems and/or virtual networks. Students who are already familiar with the following topics can skip the Mukyu (1R) course and proceed to the Shodan (1D) material:

- Virtual Machines and Labs
- Operating System Commands
- Programming Languages

This document is a companion guide to the “Virtual Machines and Labs” portion of the Mukyu (1R) course. For instructional material related to the other topics, please visit the Student Area at <http://hackingdojo.com/wiki/>

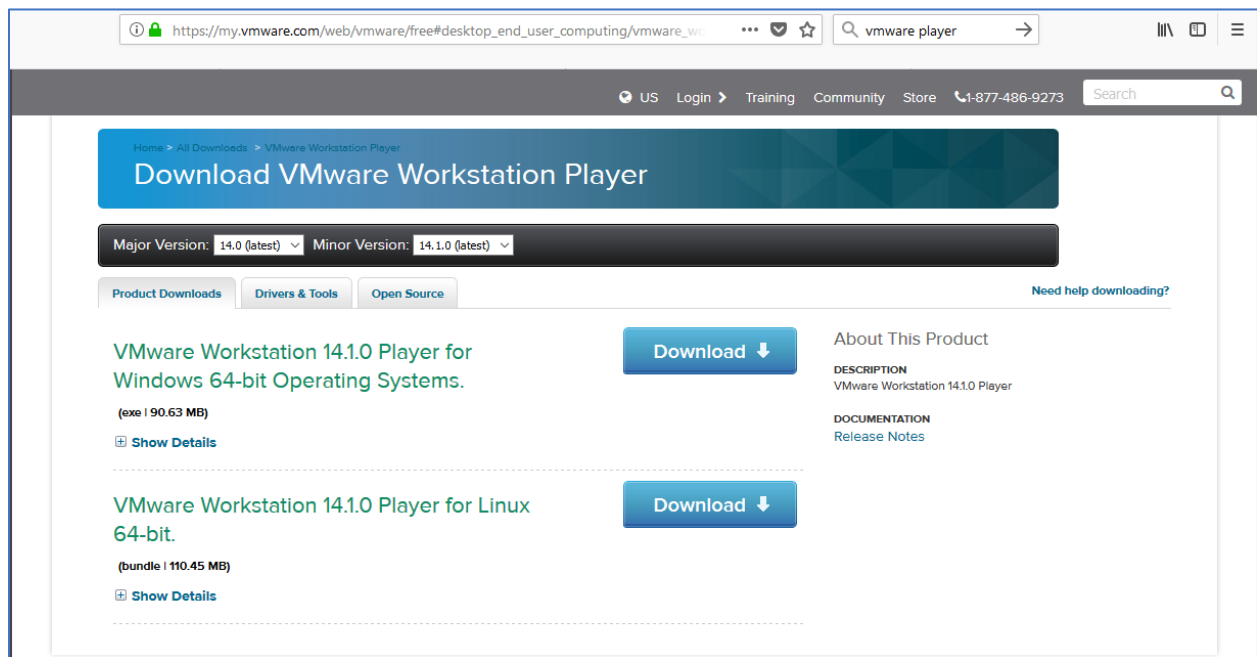
2.1 INSTALLATION

To begin, we need something that will allow us to host server images virtually on our local machine. The example I will use here is VMware Workstation Player. To obtain the application, you will need to visit the VMware site and search for their download page. At the time of writing, you can download the application at:

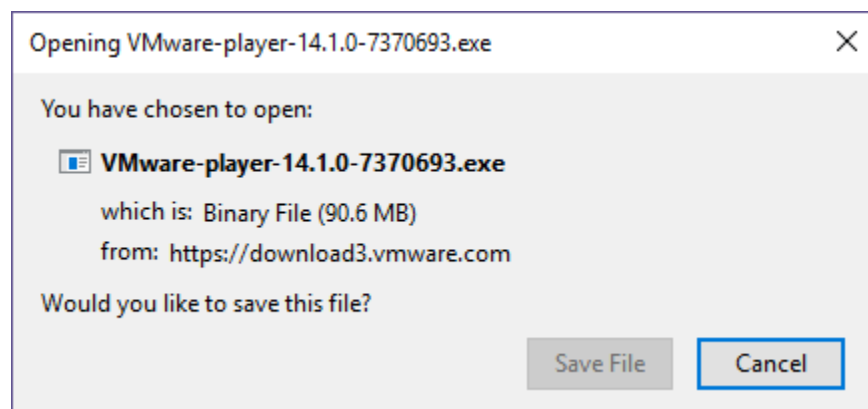
https://my.vmware.com/web/vmware/free#desktop_end_user_computing/vmware_workstation_player/14_0

Honestly, the best way to find it is to search for it on their site. The actual download location changes frequently, so I'll apologize in advance if the link above doesn't work.

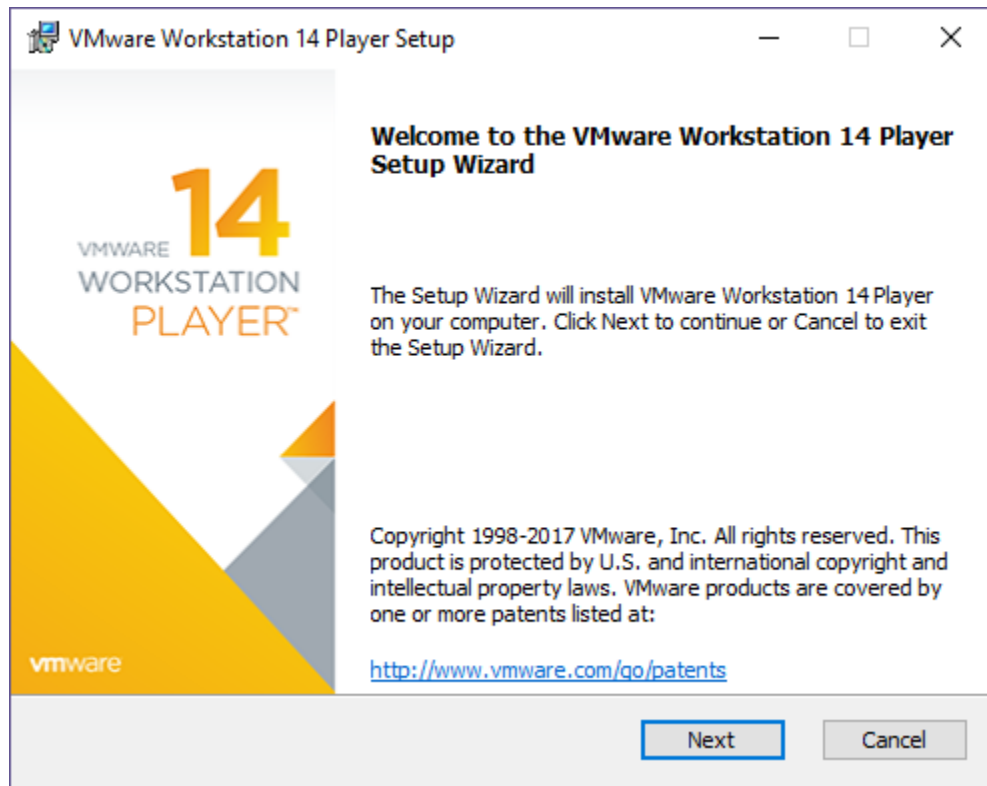
The download page is shown below. You will want to get the application that works on your platform. For this example, we selected the Windows 64-bit.



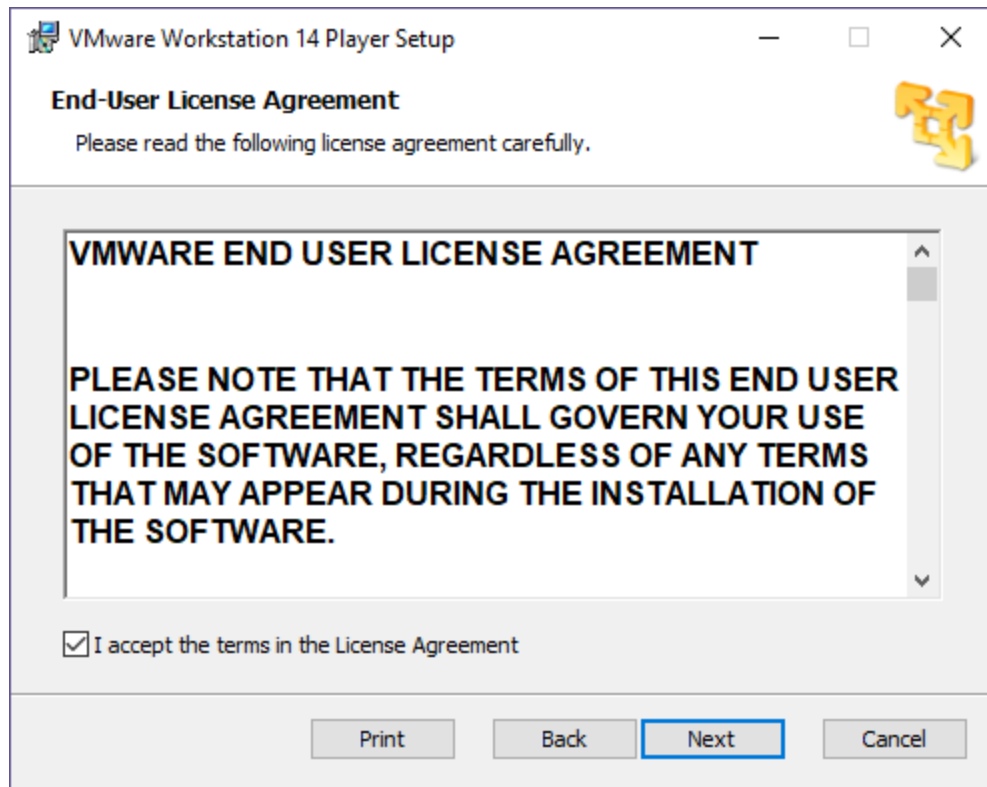
We saved the file to our local download folder as seen below:



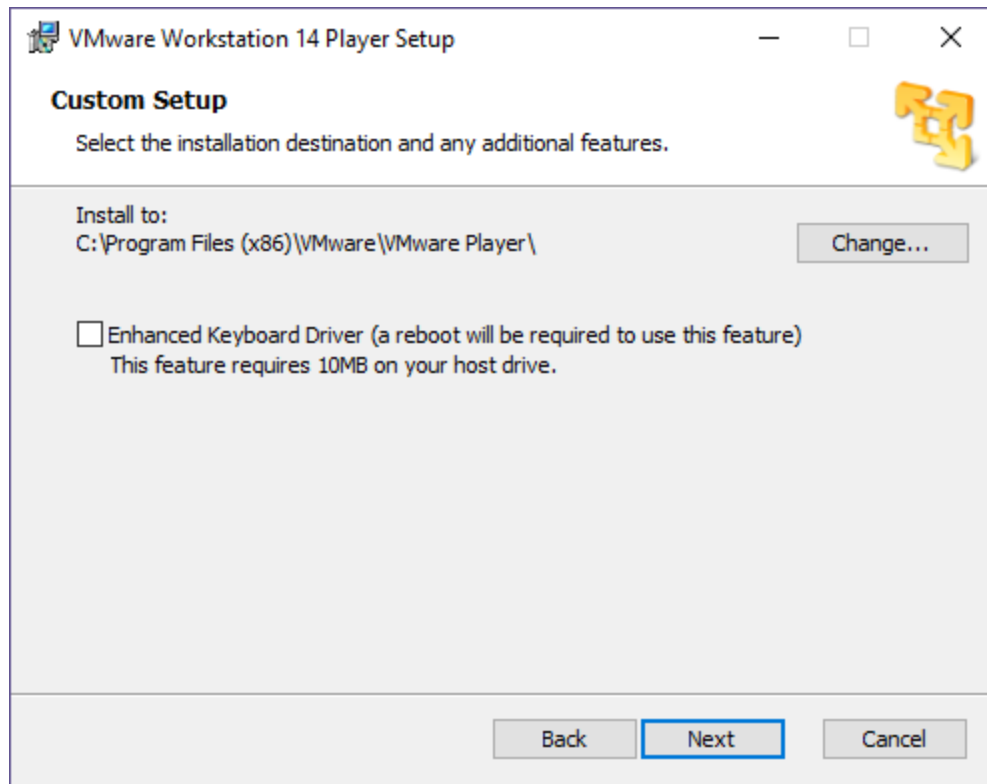
Once the file downloaded, we ran the installation application and were presented with the following screen (we will walk through each step, just to make sure you don't have any hiccups with the install):



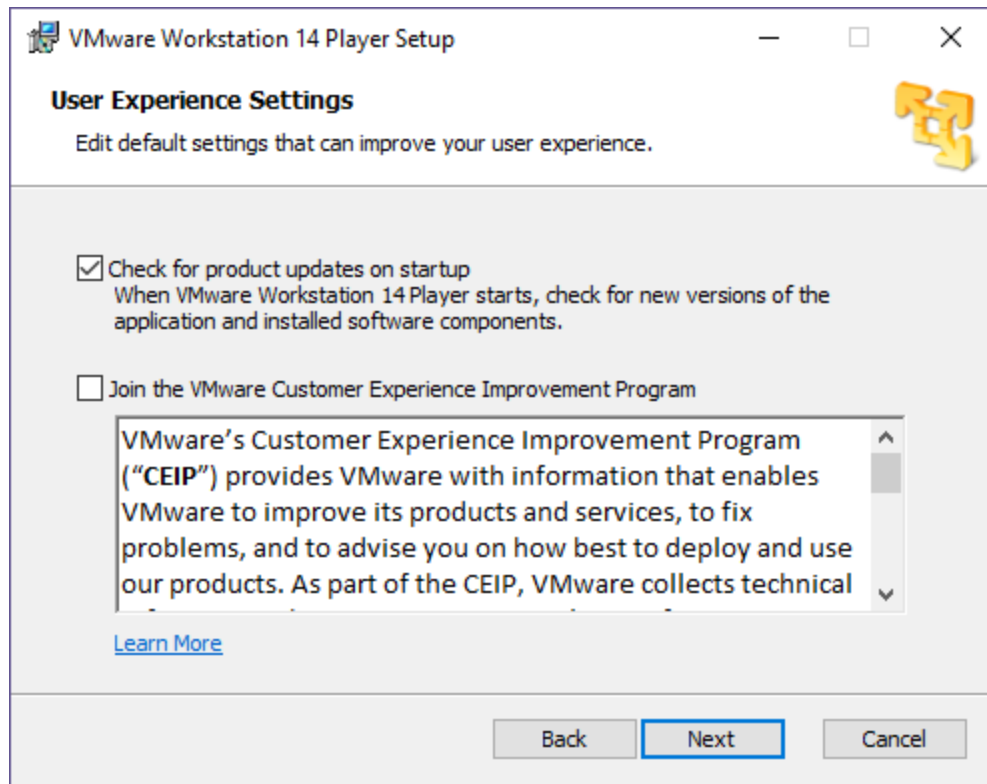
Read and accept the License Agreement:



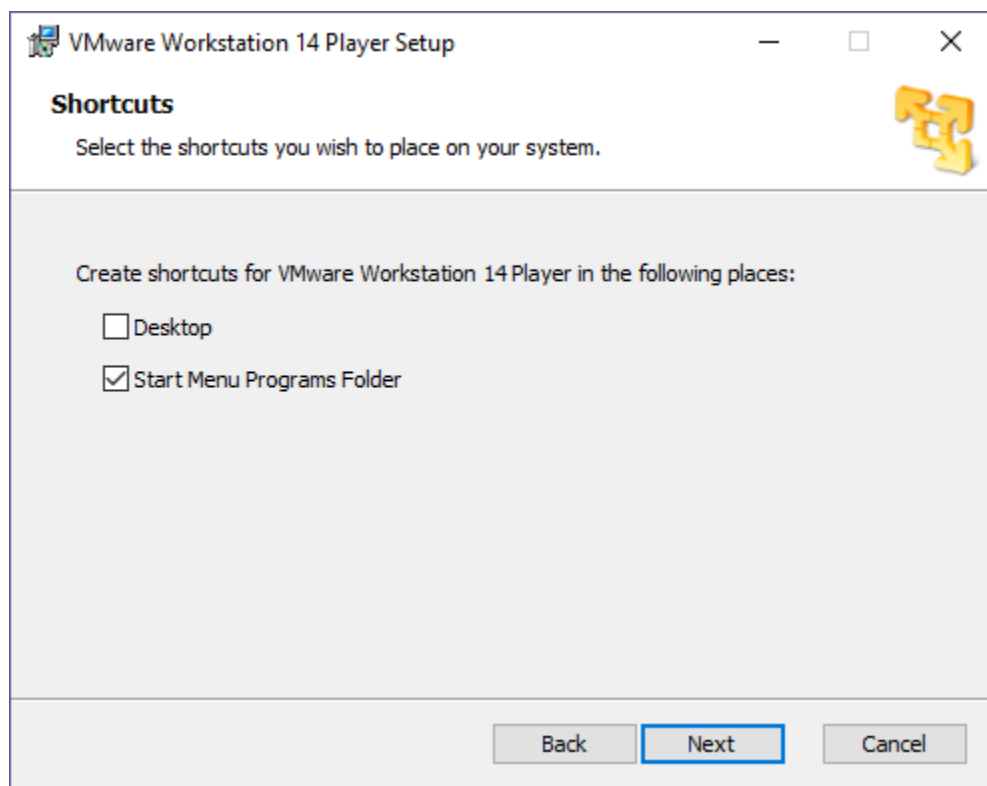
You can select “Enhanced Keyboard Driver” if you prefer, but in our example we left it blank and selected “next”:



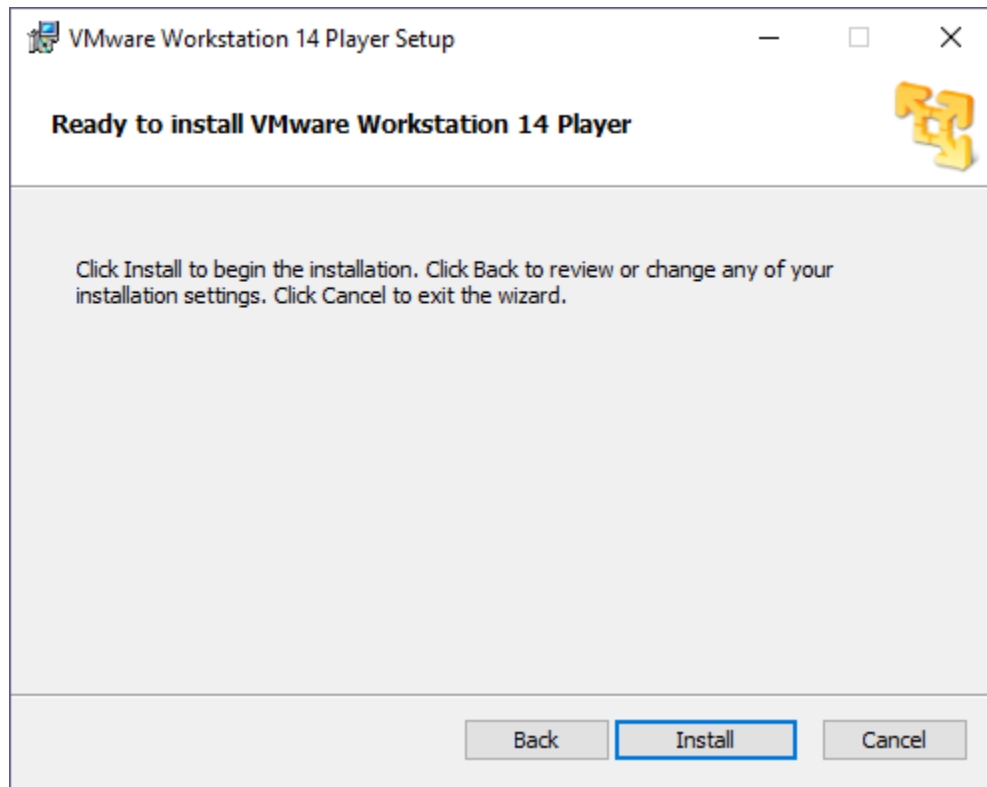
The next two options are purely up to you – I selected to allow update checking upon startup for convenience sake:



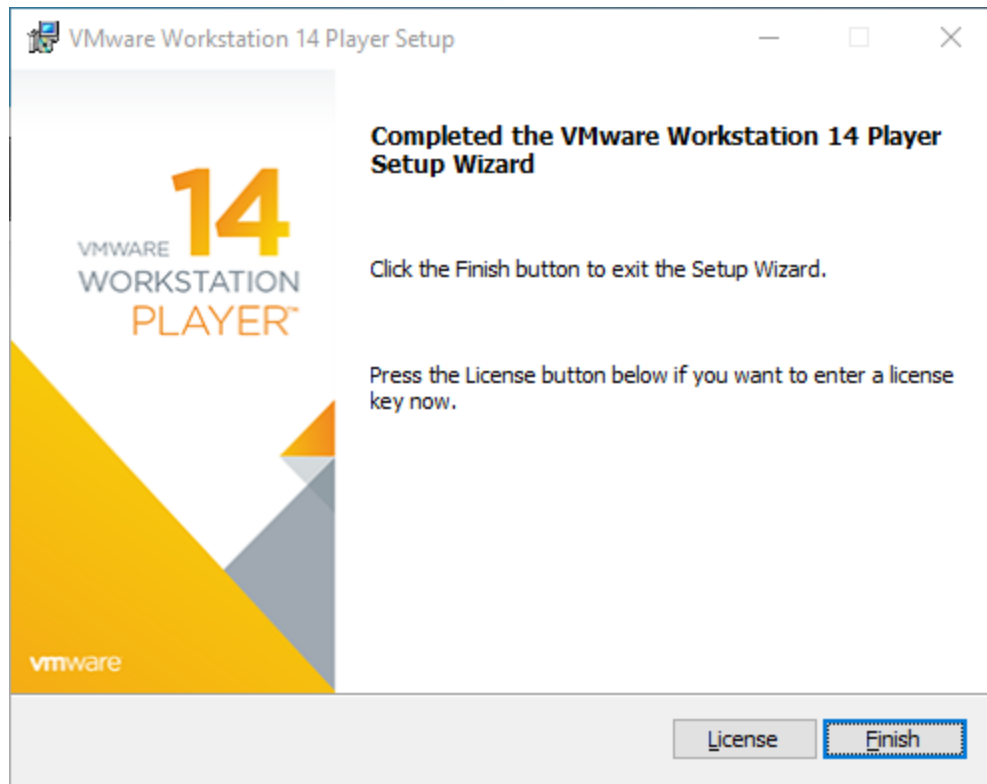
Again, choices are entirely up to you on the next screen:



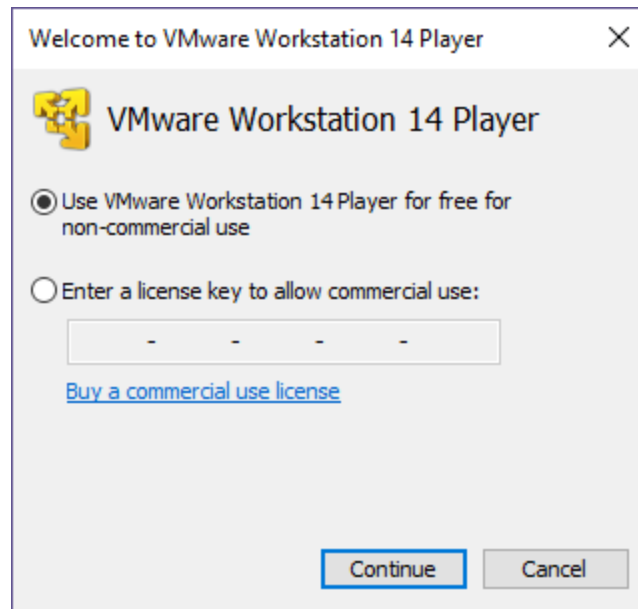
...and hit “Install” to finish this portion up:



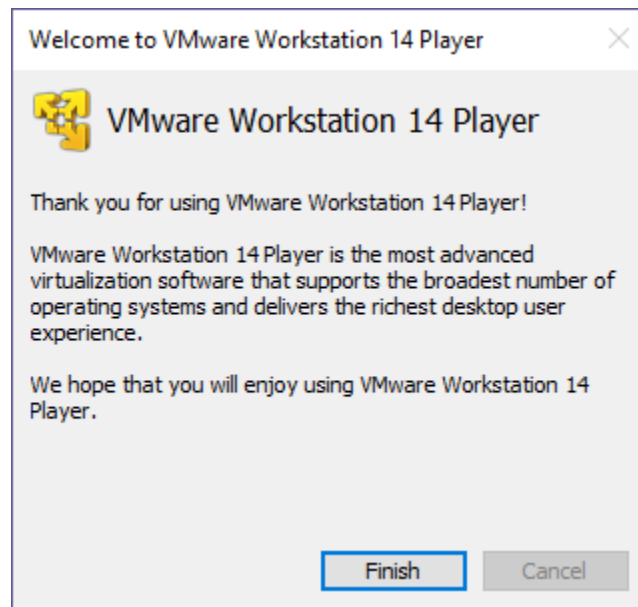
Eventually you'll be presented with the following screen, which will mean you successfully installed the VMware Workstation Player:



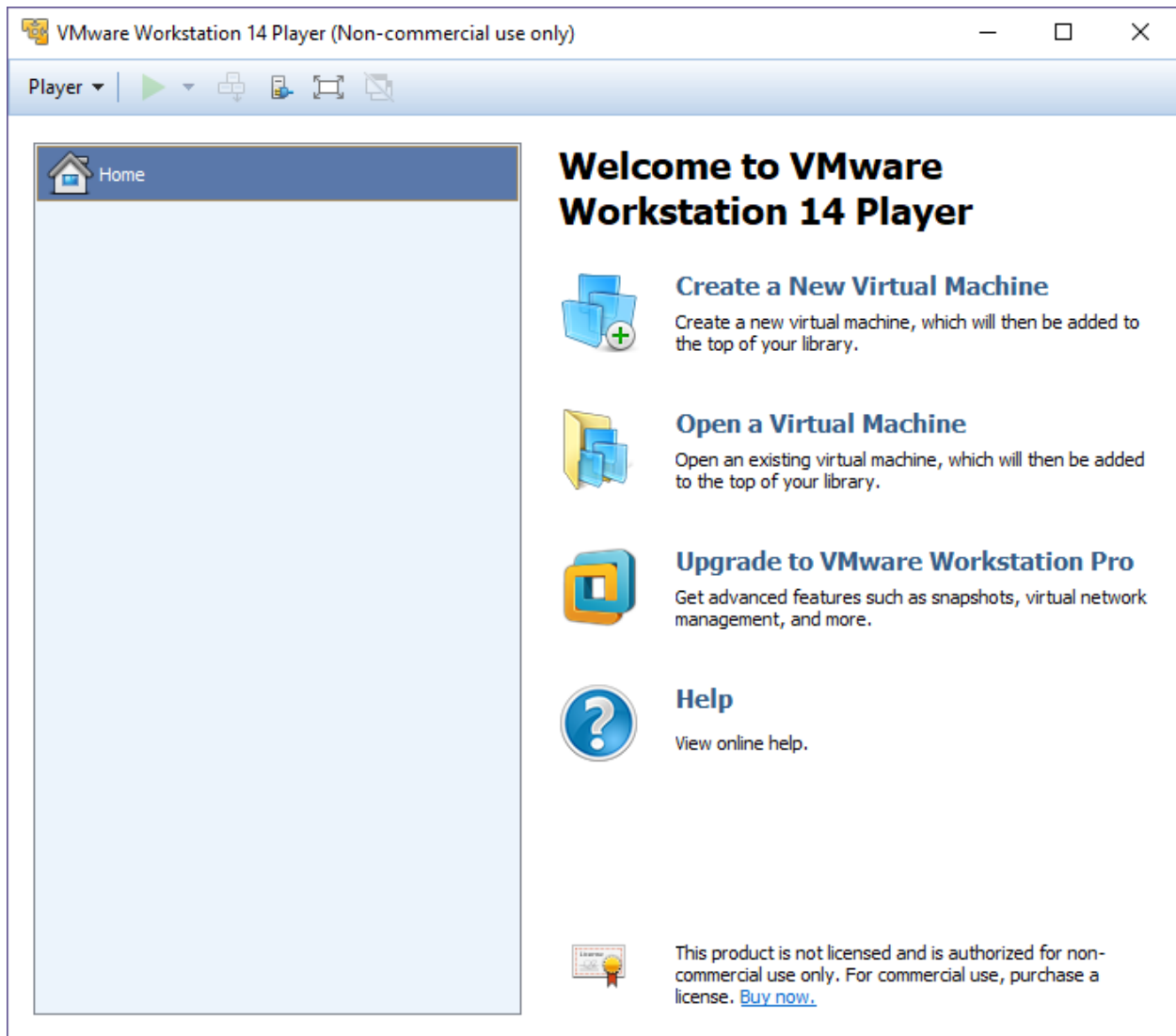
Once you have VMware Workstation Player installed, launch the application. Once started, you will be presented with the following window – select the appropriate choice.



Once your choice has been selected and you hit “Continue” you will receive a prompt that you have completed installation and licensing:



You should now have the following window appear – this will be your starting point for installation and running all the virtual images used in Hacking Dojo exercises when you aren’t using the lab:



That's all there is to installing and configuring the VMware Workstation Player!

2.2 INSTALLING KALI

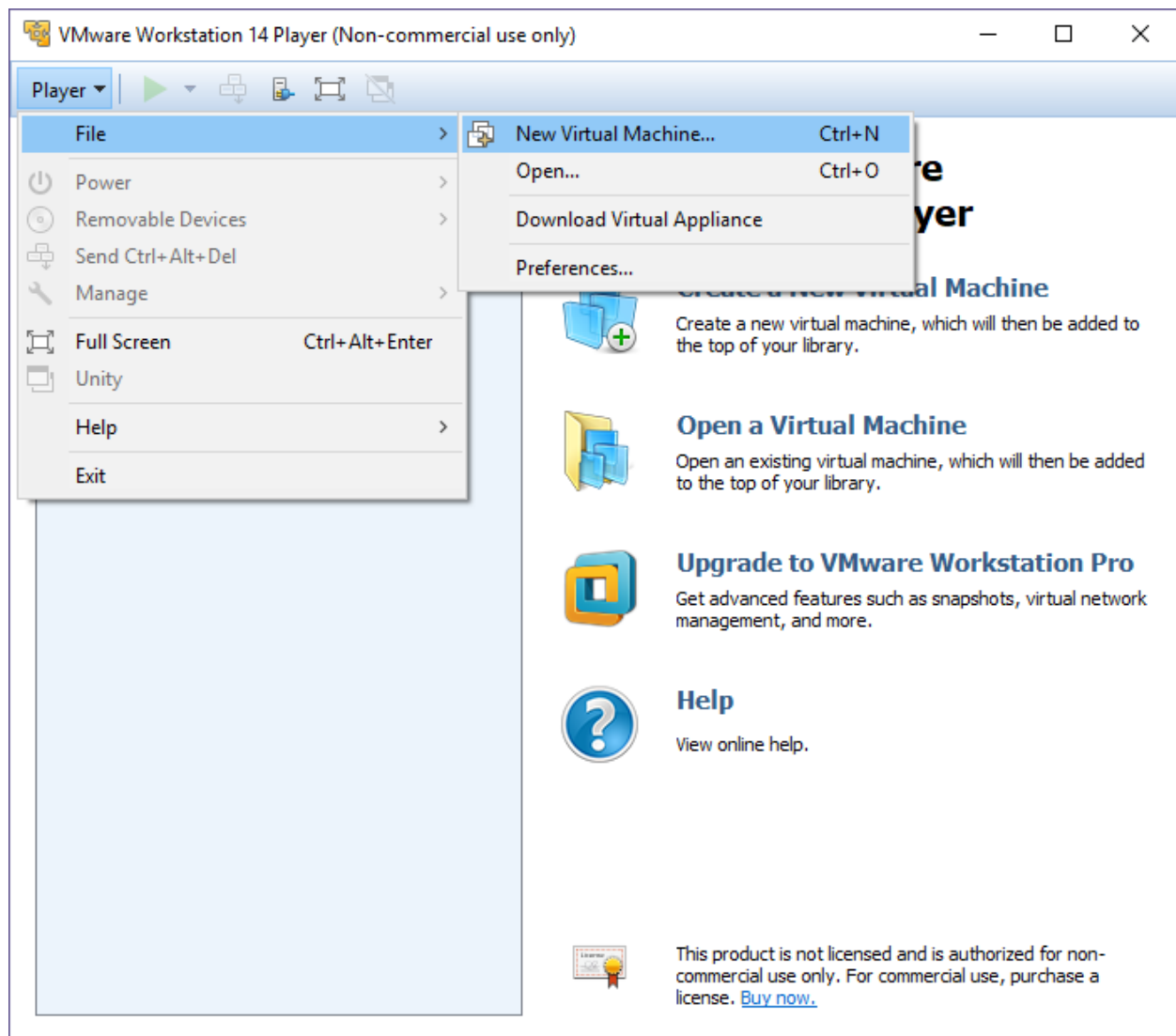
The next step we will take is to install the Kali Linux distro. To obtain the latest version of Kali, visit the following link and download the Kali 64 bit version:

<https://www.kali.org/downloads/>

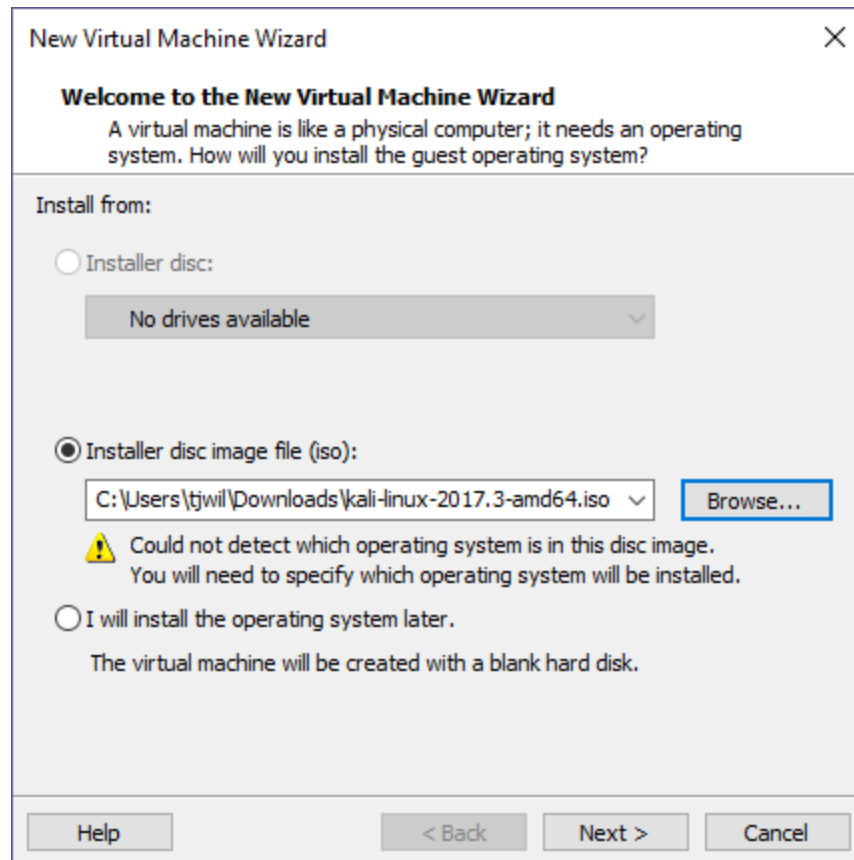
When completed, you will have an ISO file, similar to the one shown below:

Name	Date modified	Type	Size
 kali-linux-2017.3-amd64	1/6/2018 2:23 PM	Disc Image File	2,818,752 KB
 VMware-player-14.1.0-7370693	1/6/2018 2:03 PM	Application	92,809 KB

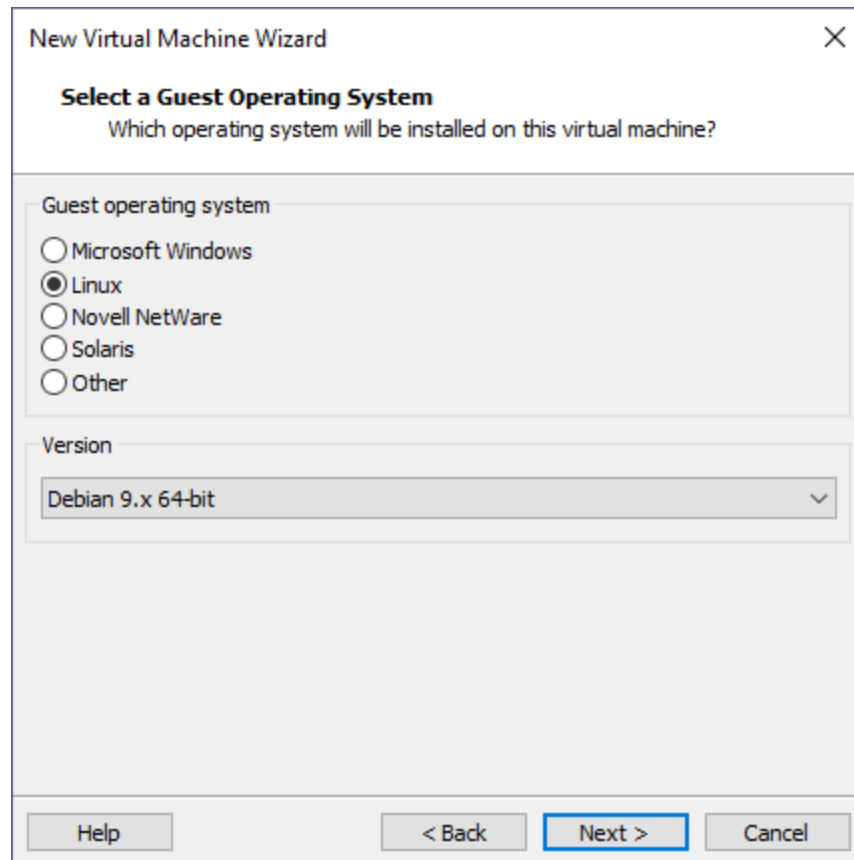
Remember where this Disc Image File is for the next step. Return to your VMware Workstation Player, and choose to create a “New Virtual Machine...” as shown below:



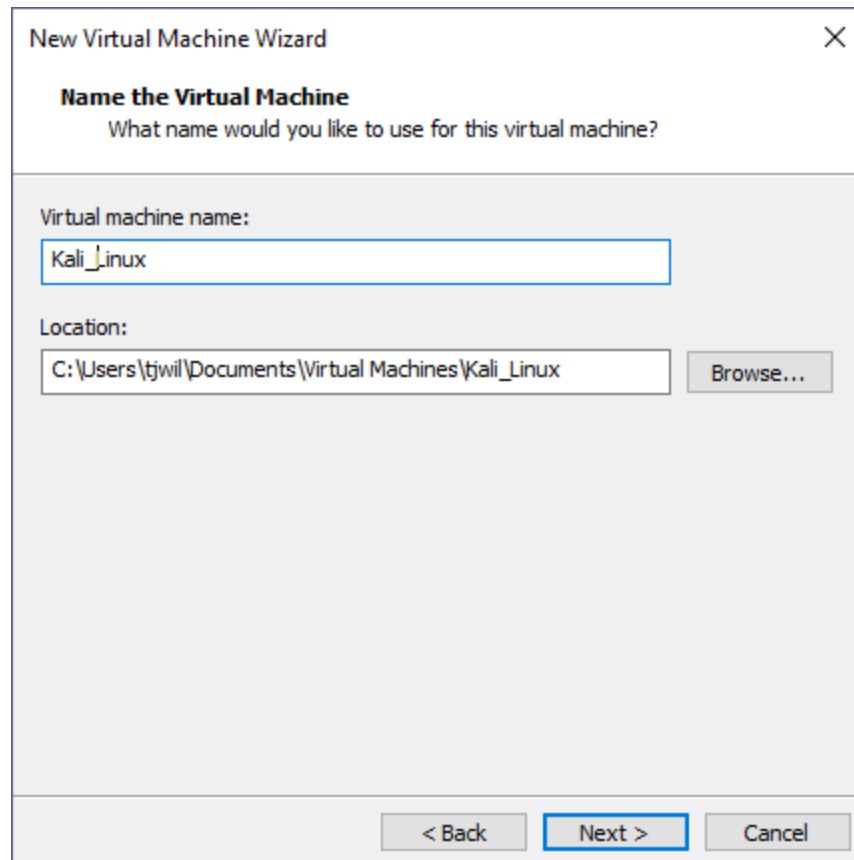
Use the “Browse” button to select the Kali Disc Image File (.iso) we downloaded, and click “Next” as shown below:



When prompted, select the Linux version that your version of Kali is using – in this example, we chose Debian 9.x 64-bit. Other than the “64-bit” part, it isn’t absolutely critical you get the correct version of the OS:



Once you select “Next” you will be asked to make it something – the “Location” will change to whatever you name the Virtual Machine in the name field, as seen here:



New Virtual Machine Wizard

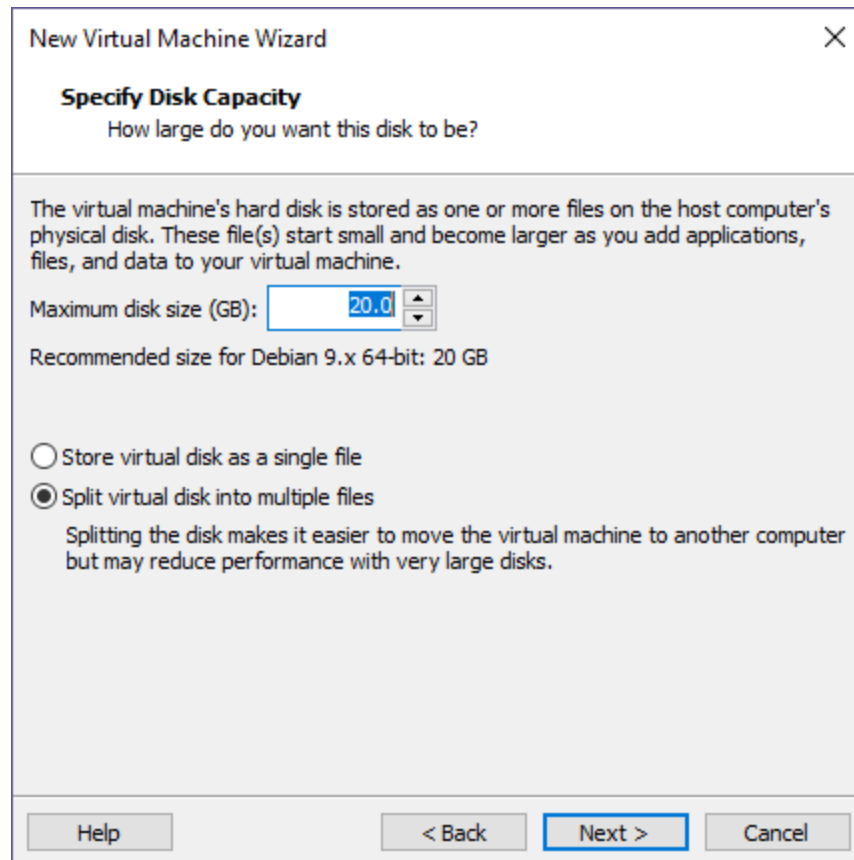
Name the Virtual Machine
What name would you like to use for this virtual machine?

Virtual machine name:
Kali_Linux

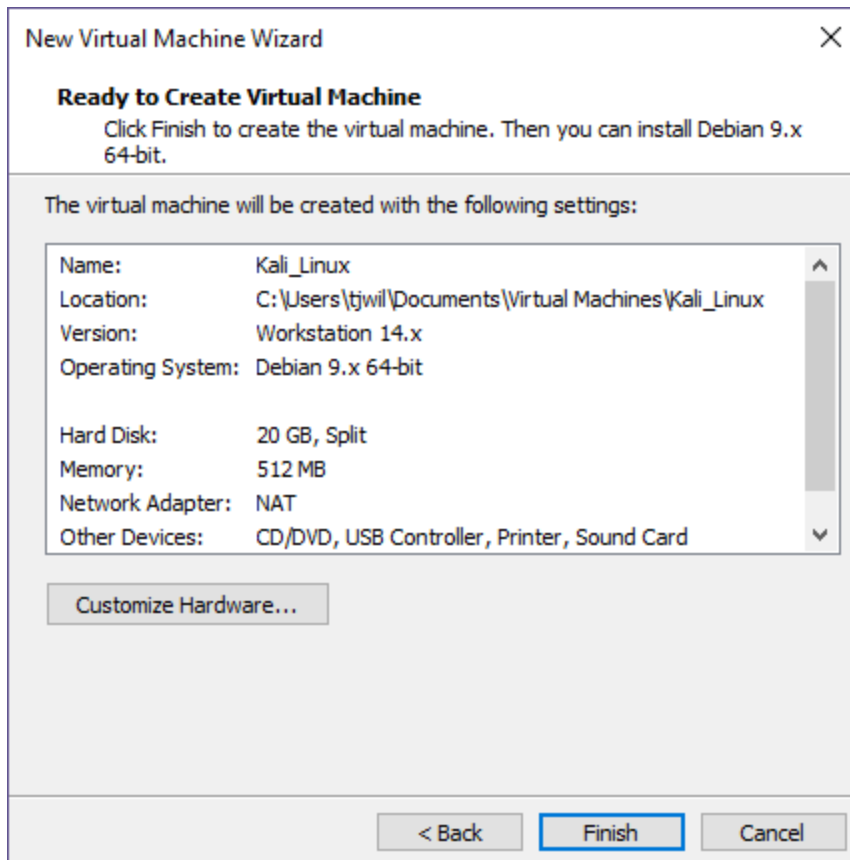
Location:
C:\Users\tjwil\Documents\Virtual Machines\Kali_Linux Browse...

< Back Next > Cancel

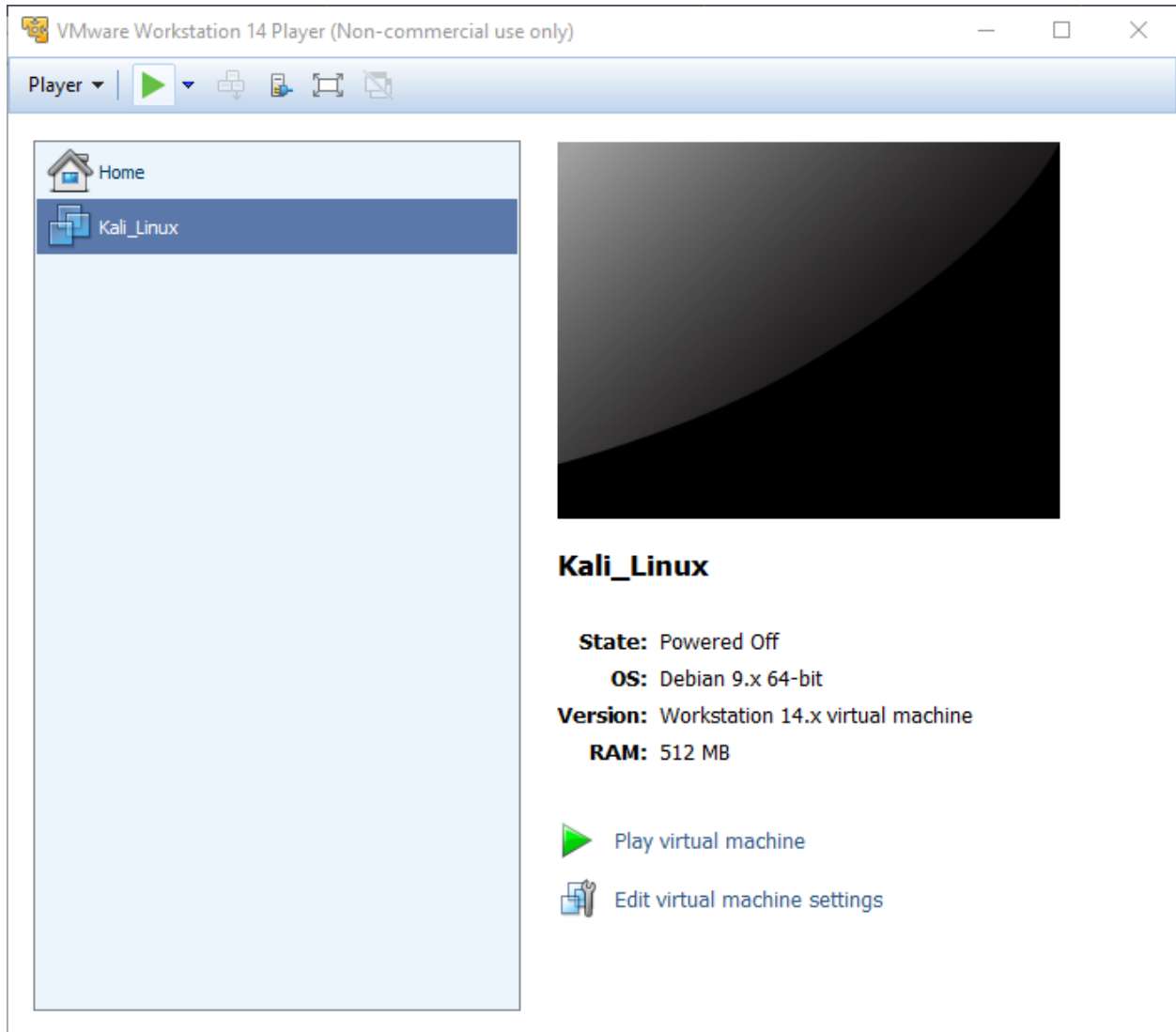
You can select the default configuration of the image, or change it as necessary for your system:



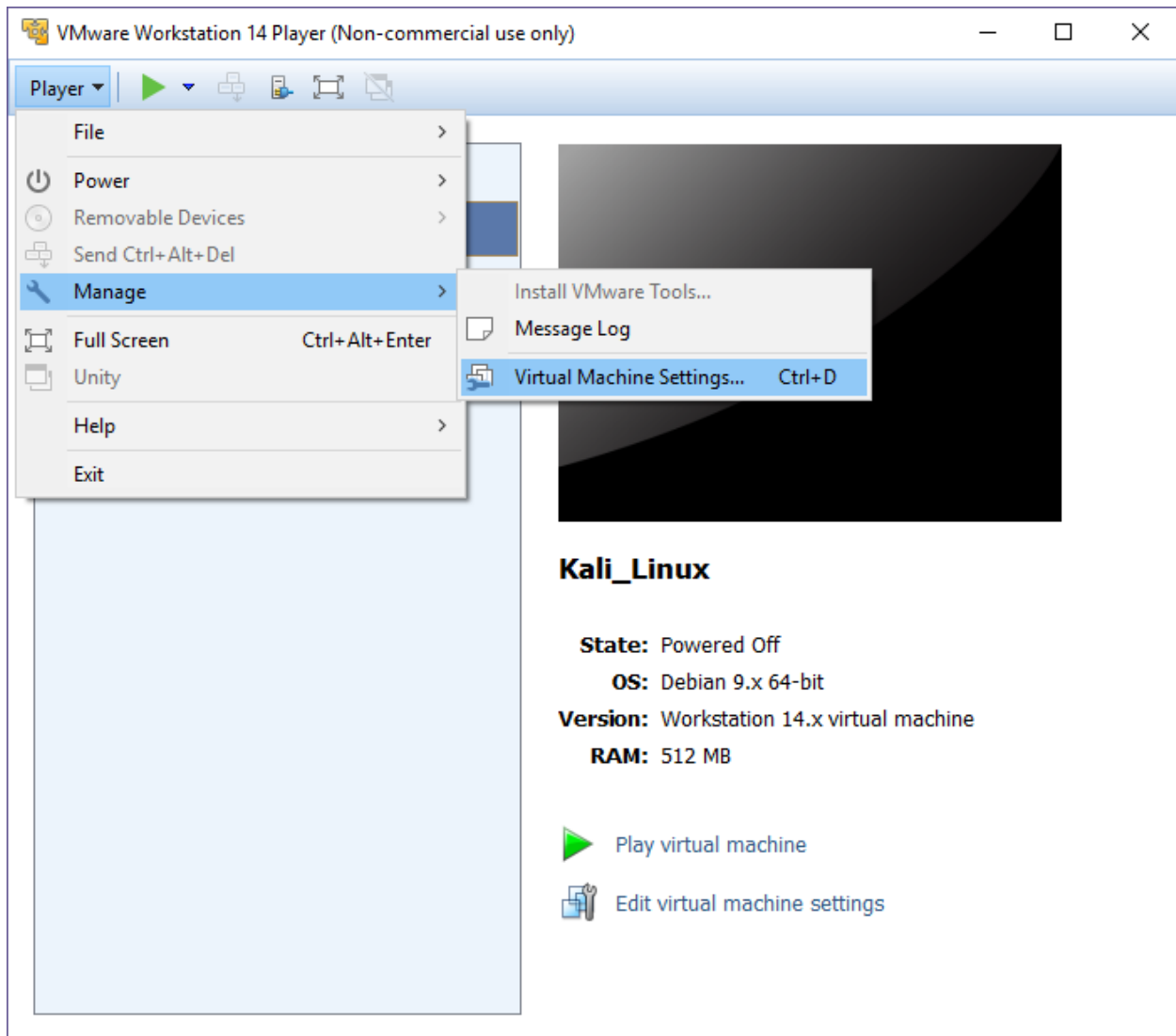
Once you hit “Next” you will be presented with the following image – click “Finish” to create your Kali Linux virtual image:



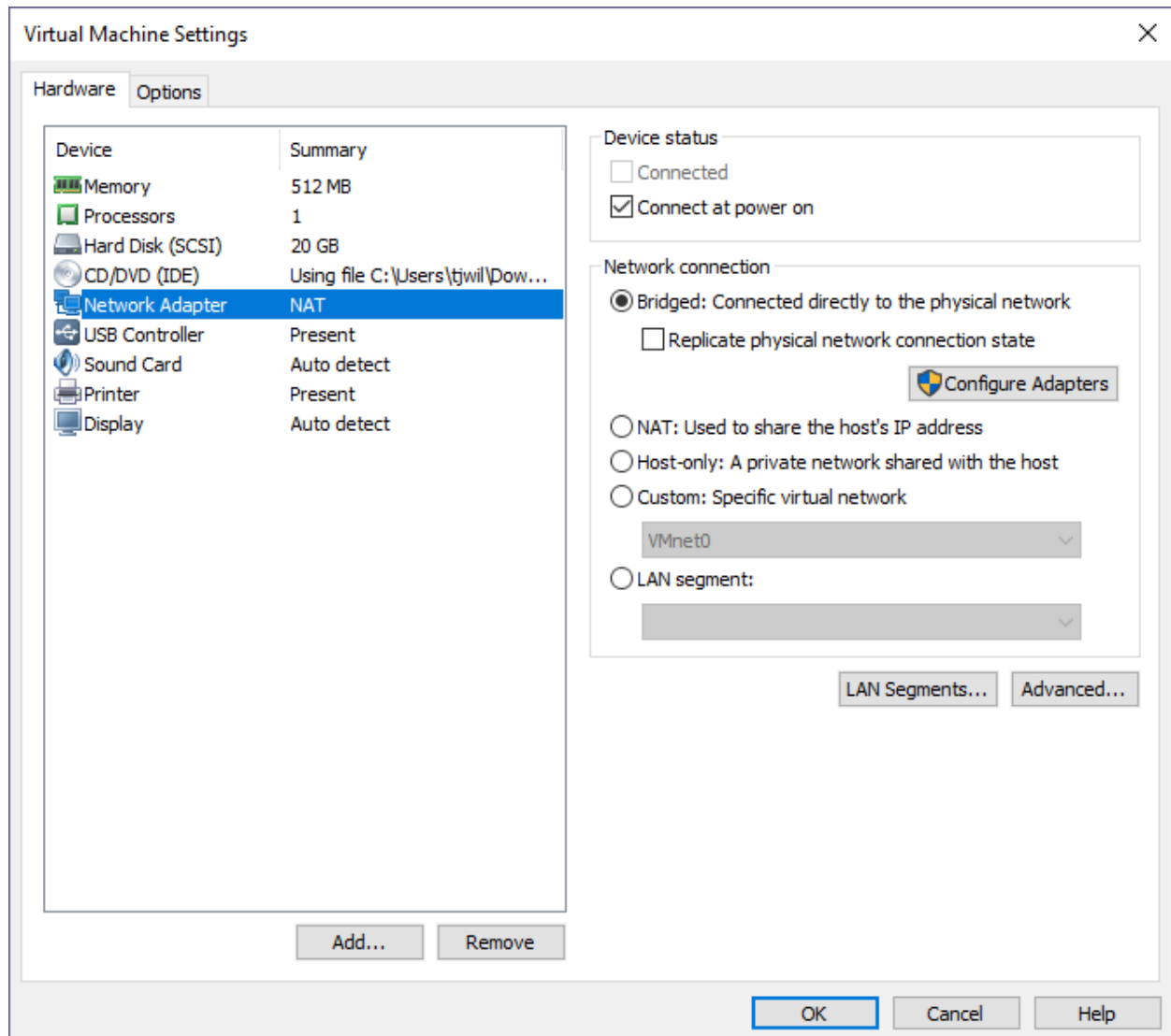
Once completed, you'll see the following on the VMware Workstation Player screen:



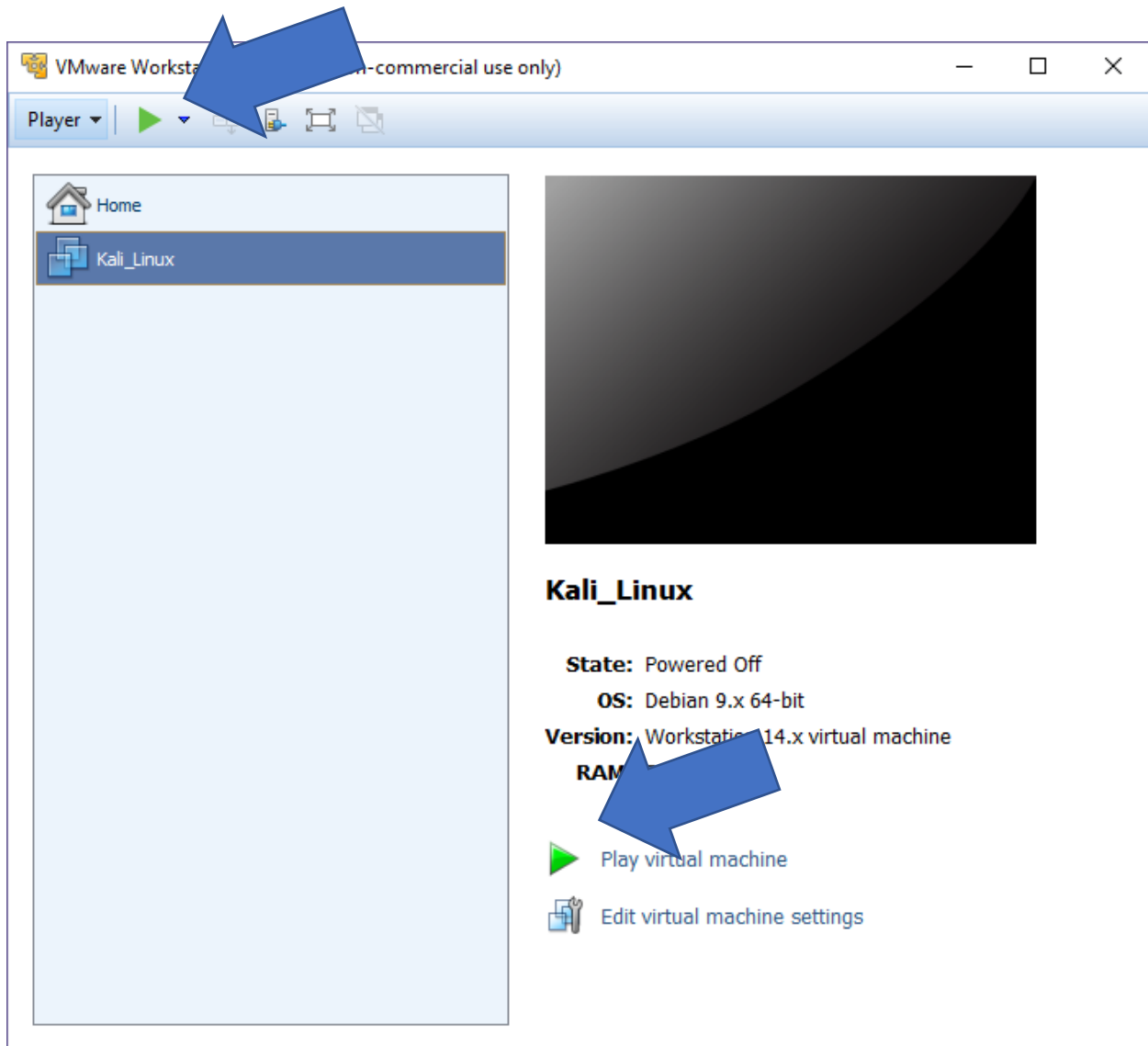
We need to modify the network configuration first before we launch it. To do so, go to the “Virtual machine Settings...” option as seen below:



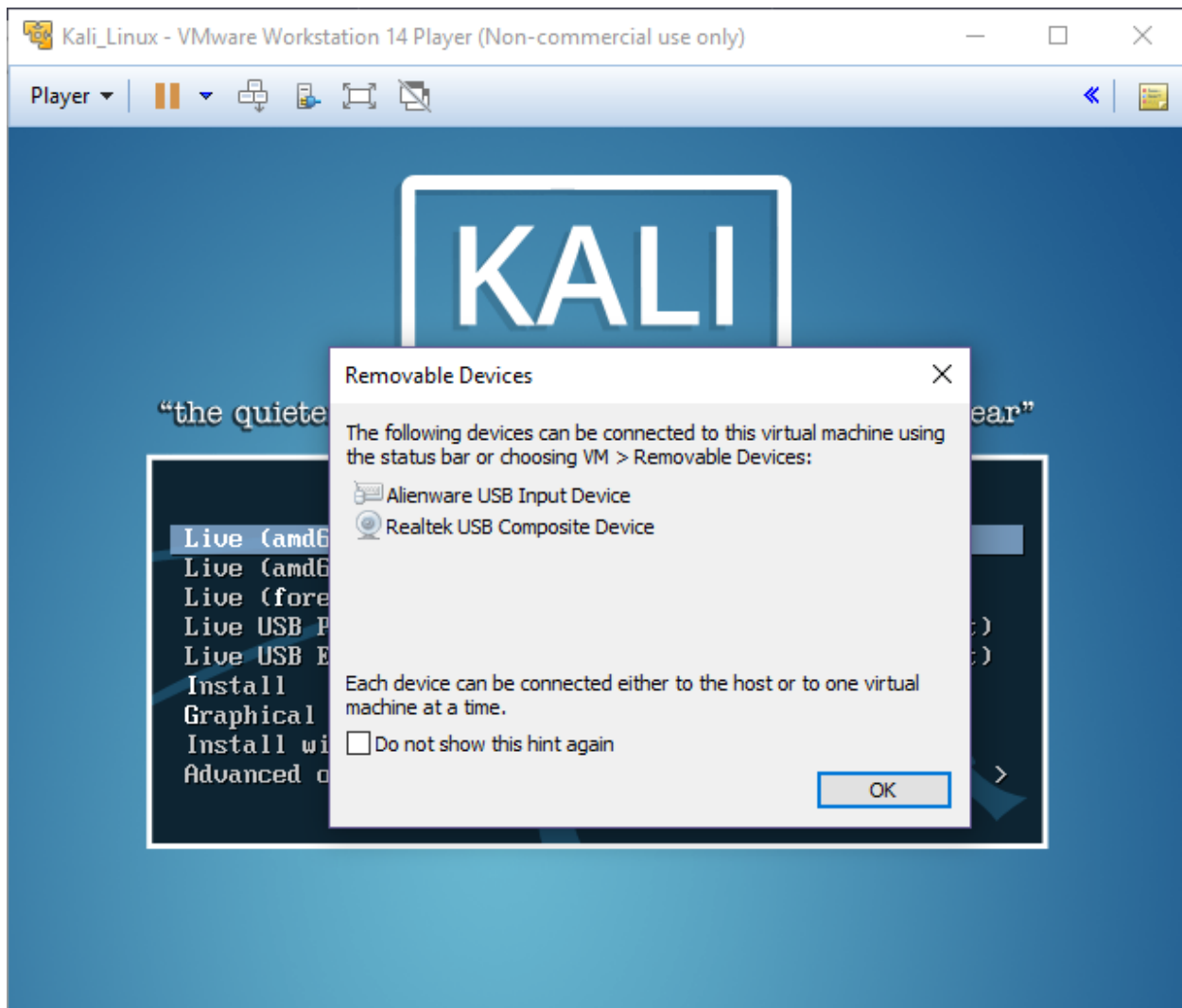
Once in the Virtual Machine Settings screen, click on “Network Adapter” and reconfigure the “Network Connection” to “Bridged” as seen in the following screen:



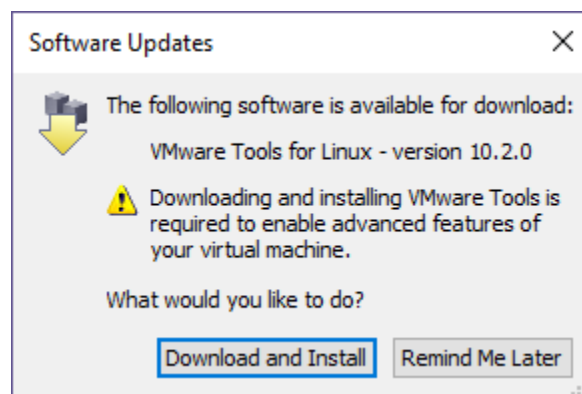
Click “OK,” return to the previous screen, and “Play virtual machine” – you can do this by hitting the Green Play button at the top of the screen, or in the right half of the screen (shown below):



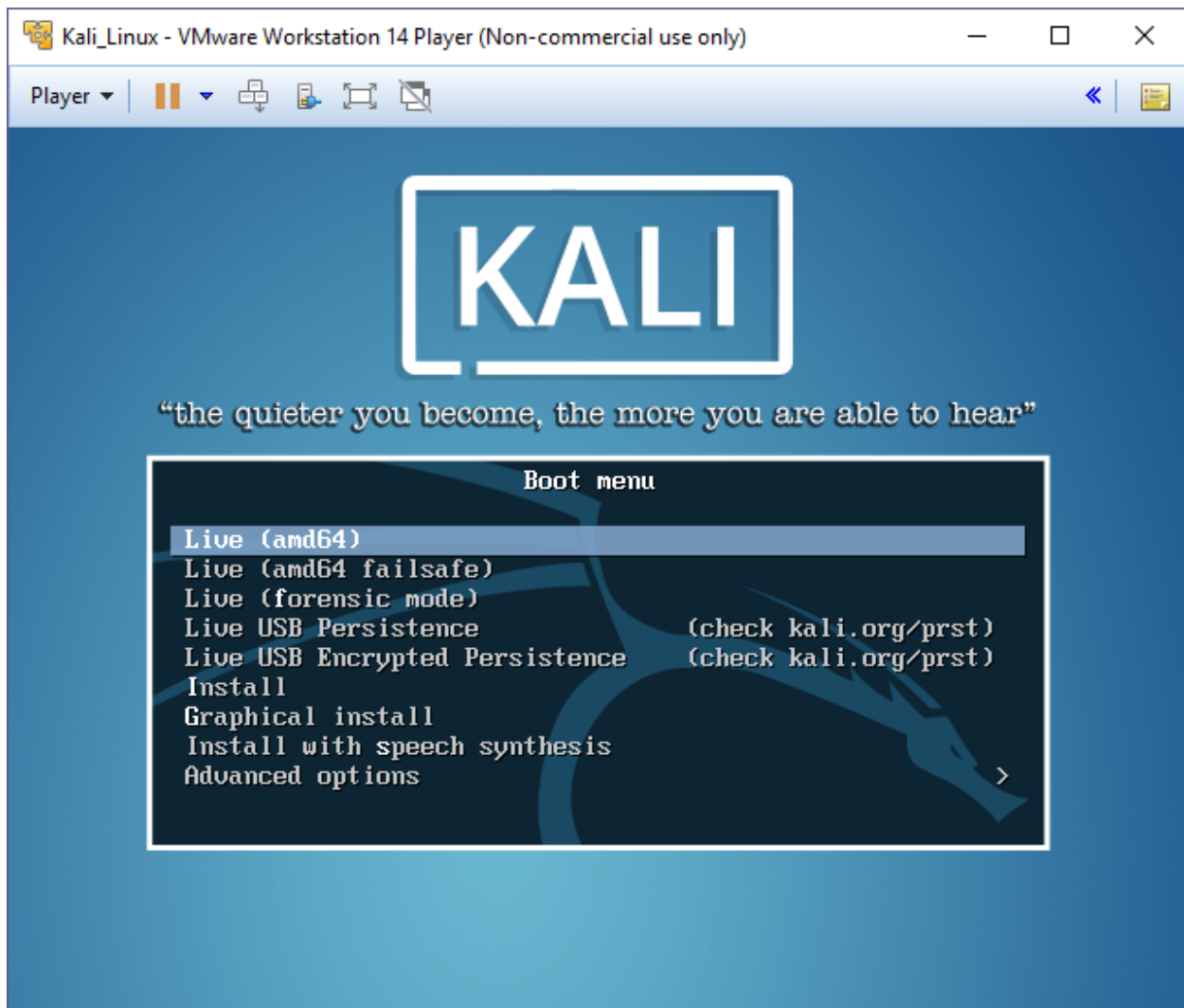
You can select the “Do not show this hint again” when prompted for removable devices – such items can be added or removed as needed in the future:



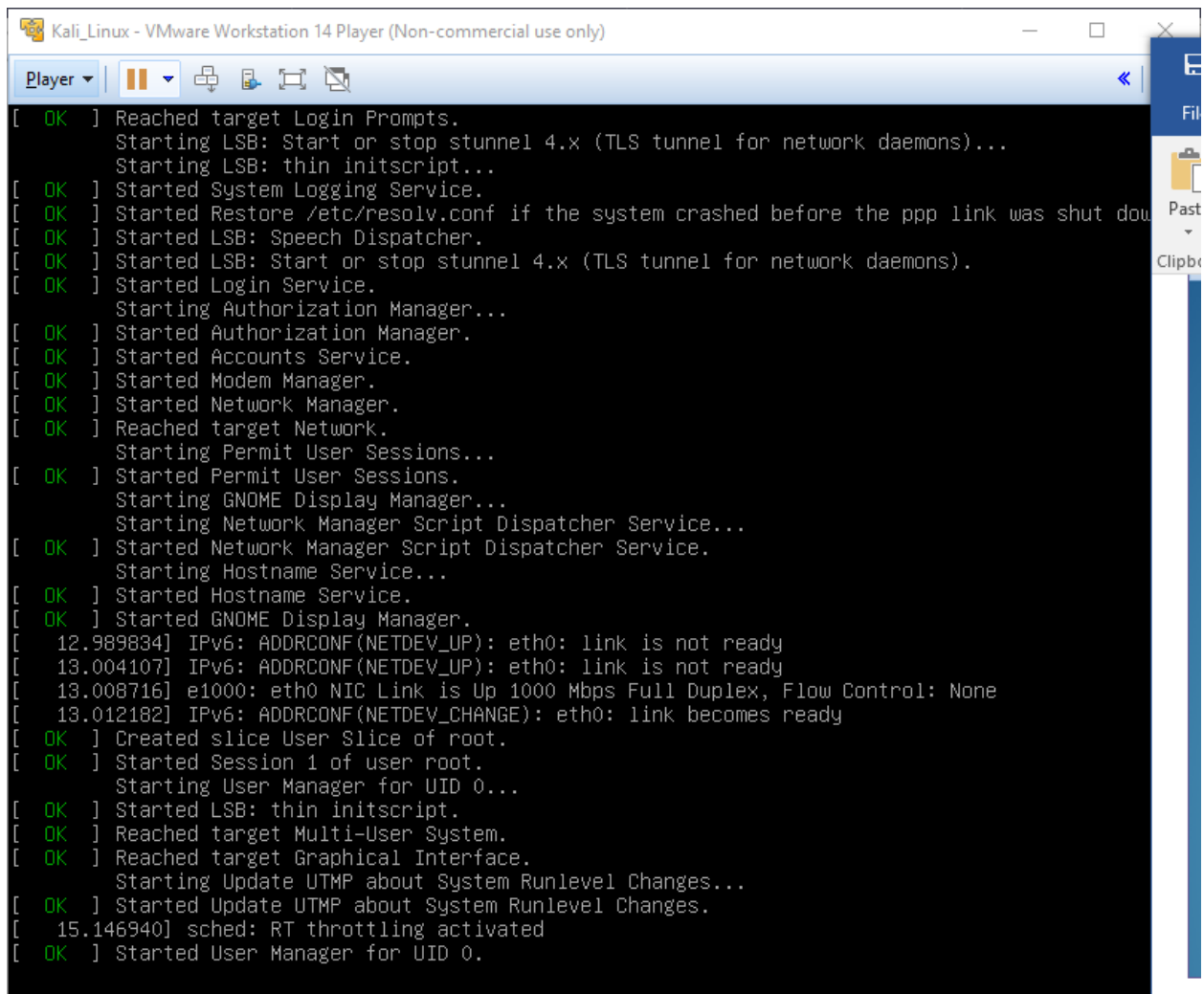
If prompted for software updates (as seen below), select your preference:



At this point, you can either install the system, or just use Kali as a Live image. In this example, since we can save our system whenever we want and for sake of expediency, we will select “Live” – however, if this is a system you intend to use regularly, I would suggest installing it instead:



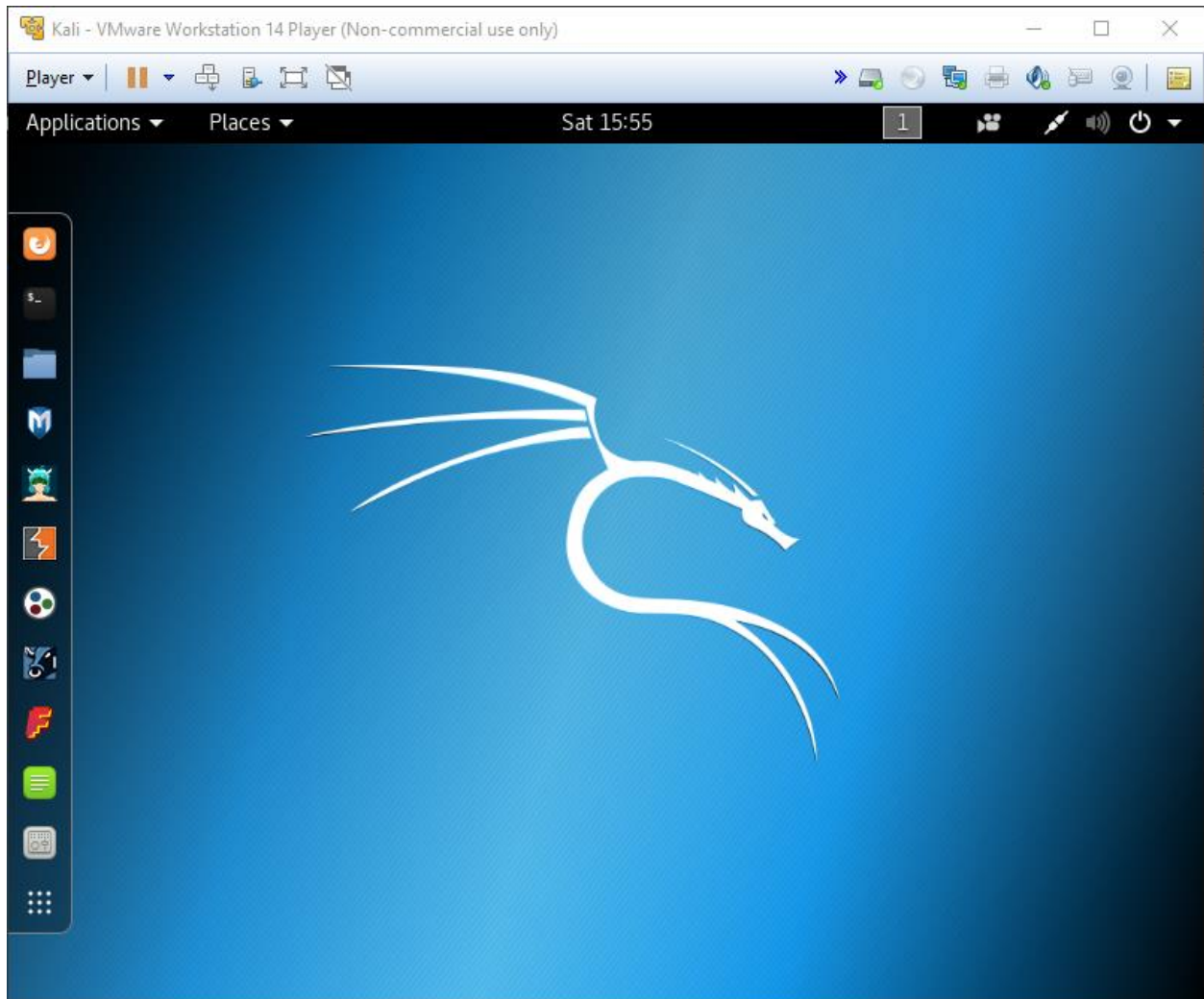
To interact with the installation, you will need to mouse over the image and left click anywhere. This will move you mouse into the virtual image and allow you to select the Live option. To return to your main screen, you will need to hit <CTRL>+<ALT>. Once you're in the virtual image screen, hit <ENTER> to launch Kali. Once started, you should see a lot of text scroll, similar to the following:



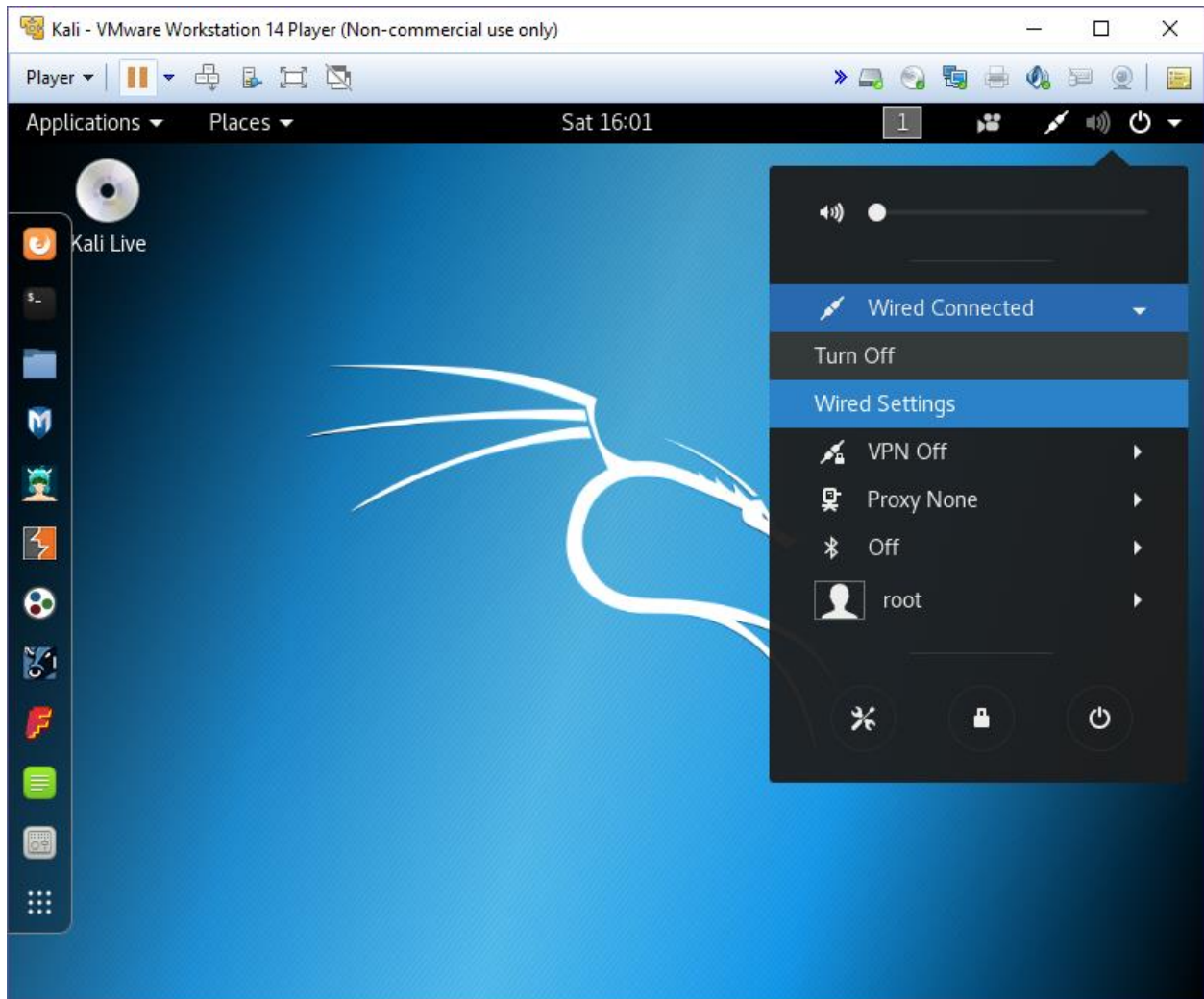
The screenshot shows a terminal window titled "Kali_Linux - VMware Workstation 14 Player (Non-commercial use only)". The terminal displays the boot sequence of a Kali Linux system. It starts with reaching the login prompts, followed by starting the LSB (Linux Service Base) and thin init script. Various system services are started, including System Logging, Restore /etc/resolv.conf, Speech Dispatcher, and Login Service. The system then starts the Authorization Manager, Accounts Service, Modem Manager, and Network Manager. It reaches the network target and starts Permit User Sessions, GNOME Display Manager, and Network Manager Script Dispatcher Service. The system then starts the Hostname Service and GNOME Display Manager. Network configuration messages are shown, including IPv6 ADDRCONF and eth0 NIC link status. The system creates a user slice, starts a session for user root, and starts the User Manager for UID 0. The system then starts the LSB thin init script, reaches the multi-user system, and starts the Update UTMP about System Runlevel Changes. Finally, the system starts the User Manager for UID 0.

```
[ OK ] Reached target Login Prompts.
Starting LSB: Start or stop stunnel 4.x (TLS tunnel for network daemons)...
Starting LSB: thin initscript...
[ OK ] Started System Logging Service.
[ OK ] Started Restore /etc/resolv.conf if the system crashed before the ppp link was shut down.
[ OK ] Started LSB: Speech Dispatcher.
[ OK ] Started LSB: Start or stop stunnel 4.x (TLS tunnel for network daemons).
[ OK ] Started Login Service.
Starting Authorization Manager...
[ OK ] Started Authorization Manager.
[ OK ] Started Accounts Service.
[ OK ] Started Modem Manager.
[ OK ] Started Network Manager.
[ OK ] Reached target Network.
Starting Permit User Sessions...
[ OK ] Started Permit User Sessions.
Starting GNOME Display Manager...
Starting Network Manager Script Dispatcher Service...
[ OK ] Started Network Manager Script Dispatcher Service.
Starting Hostname Service...
[ OK ] Started Hostname Service.
[ OK ] Started GNOME Display Manager.
[ 12.989834] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
[ 13.004107] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
[ 13.008716] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
[ 13.012182] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
[ OK ] Created slice User Slice of root.
[ OK ] Started Session 1 of user root.
Starting User Manager for UID 0...
[ OK ] Started LSB: thin initscript.
[ OK ] Reached target Multi-User System.
[ OK ] Reached target Graphical Interface.
Starting Update UTMP about System Runlevel Changes...
[ OK ] Started Update UTMP about System Runlevel Changes.
[ 15.146940] sched: RT throttling activated
[ OK ] Started User Manager for UID 0.
```

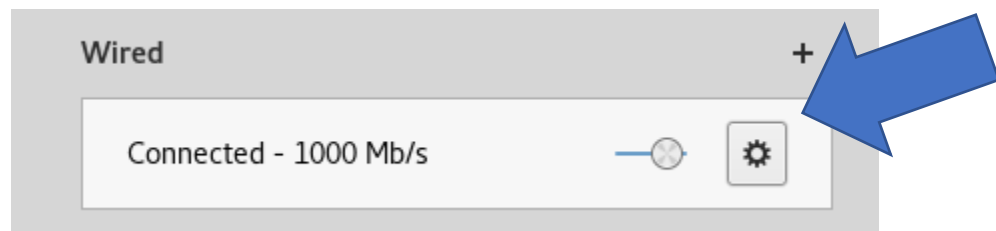
Eventually, you will be presented with the desktop for your Kali image:



At this point, we need to modify the IP address of the system to match the network of that in the lab. To do so, you will need to configure the wired settings – to do so, select “Wired Settings” as illustrated below:



Select the “gear” to configure the setting of the wired network device:

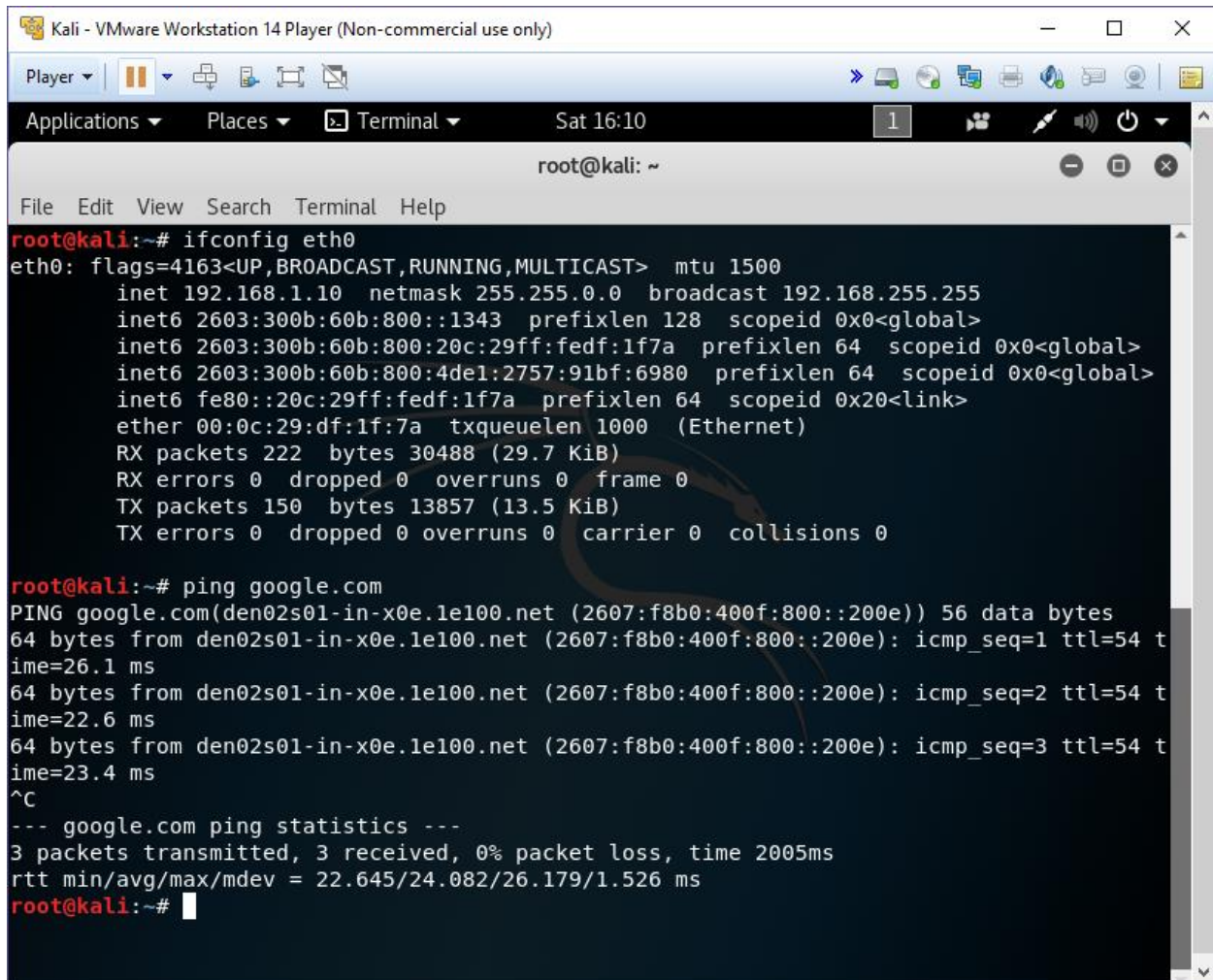


Modify the network IPv4 configuration as seen below:

The screenshot shows a network configuration window titled "Wired". It has tabs for "Details", "Identity", "IPv4", "IPv6", and "Security", with "IPv4" currently selected. At the top are "Cancel" and "Apply" buttons. Under "IPv4 Method", there are four radio buttons: "Automatic (DHCP)", "Link-Local Only", "Manual" (which is selected), and "Disable". Below this is the "Addresses" section, which contains a table with three columns: "Address", "Netmask", and "Gateway". The first row of the table contains the values "192.168.1.10", "255.255.0.0", and "192.168.0.1". A second, empty row is visible below the first. Each row has a small "x" button to its right for removal.

Address	Netmask	Gateway
192.168.1.10	255.255.0.0	192.168.0.1

At this point we have completed our configuration of the Kali image for our virtual pentest lab. To test connectivity, we can bring up a terminal window and check out network configuration using “ifconfig eth0” – which shows us that the IP address we set is correct. In addition, we can check external connectivity by pinging an external website, as shown below:



```

root@kali: ~
File Edit View Search Terminal Help
root@kali:~# ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.10 netmask 255.255.0.0 broadcast 192.168.255.255
    inet6 2603:300b:60b:800::1343 prefixlen 128 scopeid 0x0<global>
    inet6 2603:300b:60b:800:20c:29ff:fedf:1f7a prefixlen 64 scopeid 0x0<global>
    inet6 2603:300b:60b:800:4de1:2757:91bf:6980 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::20c:29ff:fedf:1f7a prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:df:1f:7a txqueuelen 1000 (Ethernet)
    RX packets 222 bytes 30488 (29.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 150 bytes 13857 (13.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~# ping google.com
PING google.com(den02s01-in-x0e.1e100.net (2607:f8b0:400f:800::200e)) 56 data bytes
64 bytes from den02s01-in-x0e.1e100.net (2607:f8b0:400f:800::200e): icmp_seq=1 ttl=54 t
ime=26.1 ms
64 bytes from den02s01-in-x0e.1e100.net (2607:f8b0:400f:800::200e): icmp_seq=2 ttl=54 t
ime=22.6 ms
64 bytes from den02s01-in-x0e.1e100.net (2607:f8b0:400f:800::200e): icmp_seq=3 ttl=54 t
ime=23.4 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 22.645/24.082/26.179/1.526 ms
root@kali:~#

```

Everything seems to be working, so let's move onto some exercises.

3 PENTEST TARGET

3.1 DE-ICE 1.100

While the Mukyu course isn't intended to actually perform any attacks against remote systems, it is valuable to at least verify that our virtual lab will work when we move onto the Shodan (1D) course. To do this, we will install an additional system that will allow us to test our configuration. To begin, we will use the De-ICE 1.100 image. This ISO was created back in 2006, and was the first LiveCD

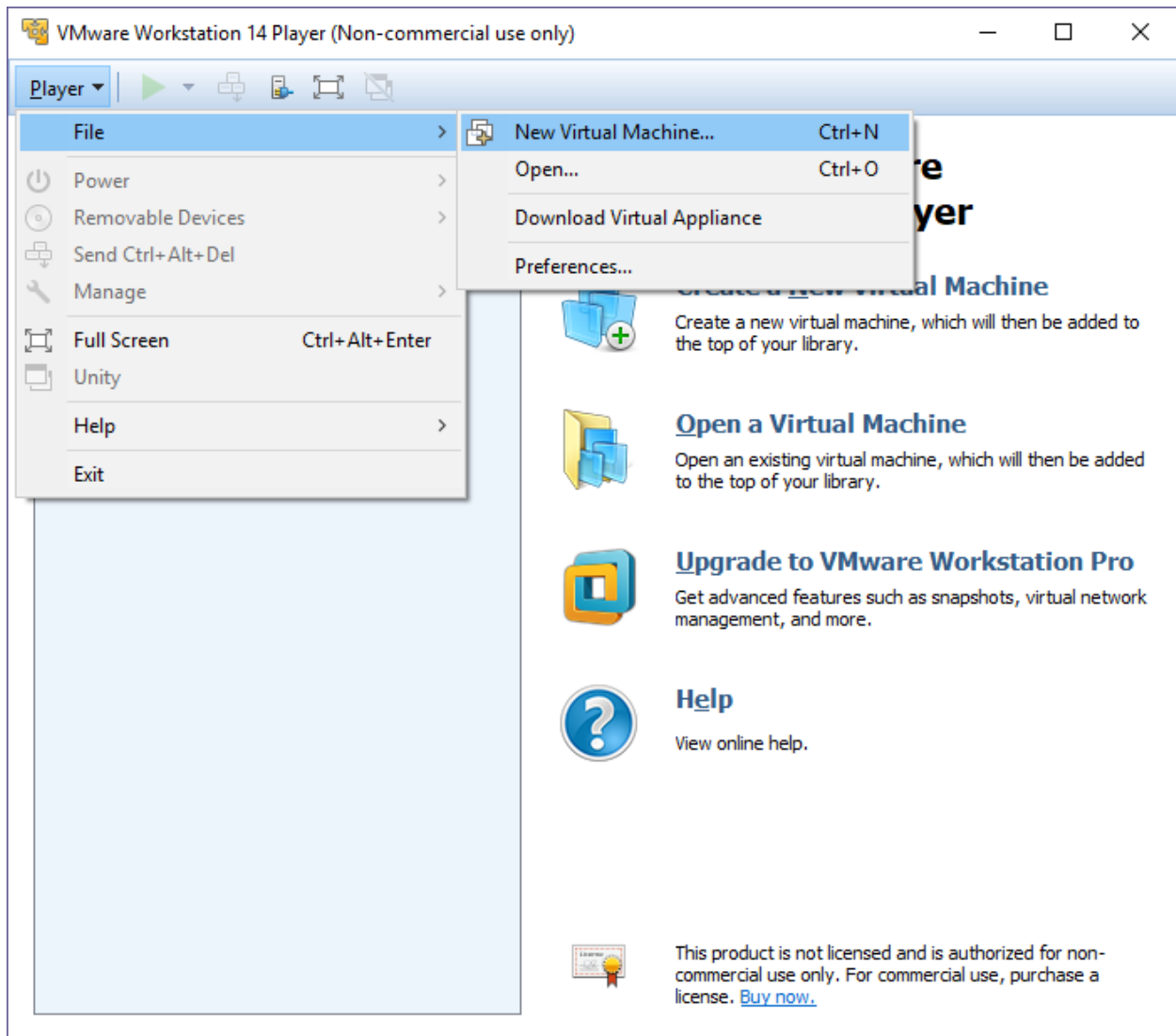
released that was intended to be exploited (for a history lesson and see me talk about the release of the De-ICE project, check out my DefCon 15 speech:

<https://www.youtube.com/watch?v=buLIGYsBCPY&t=1085s>).

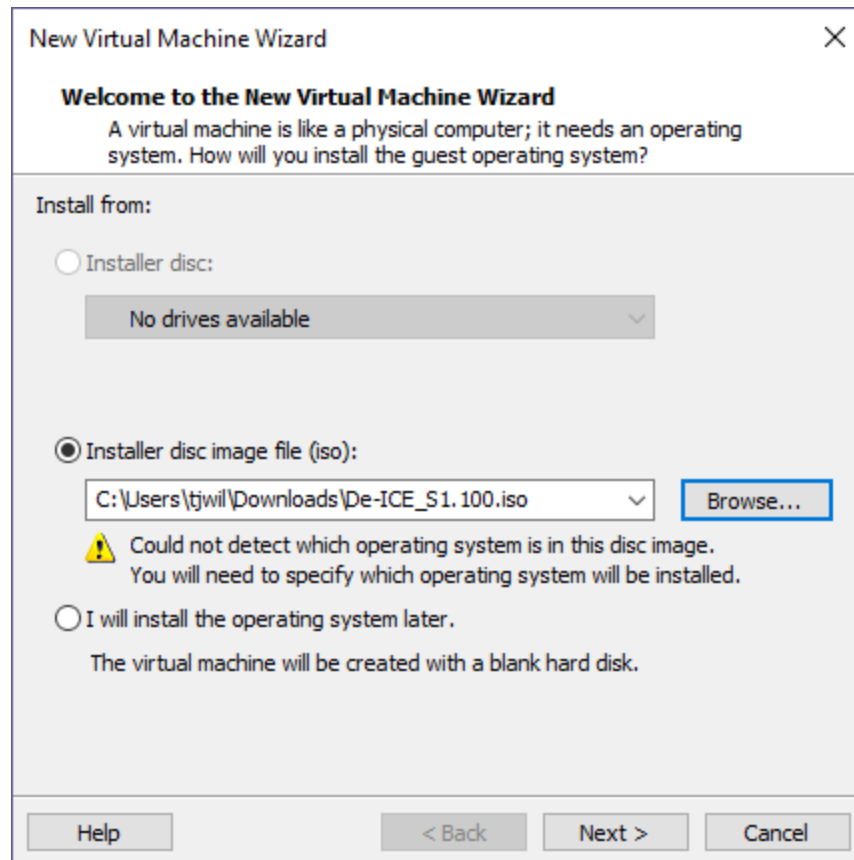
To download the ISO image, we can download it directly from the Hacking Dojo website at:

http://hackingdojo.com/downloads/iso/De-ICE_S1.100.iso

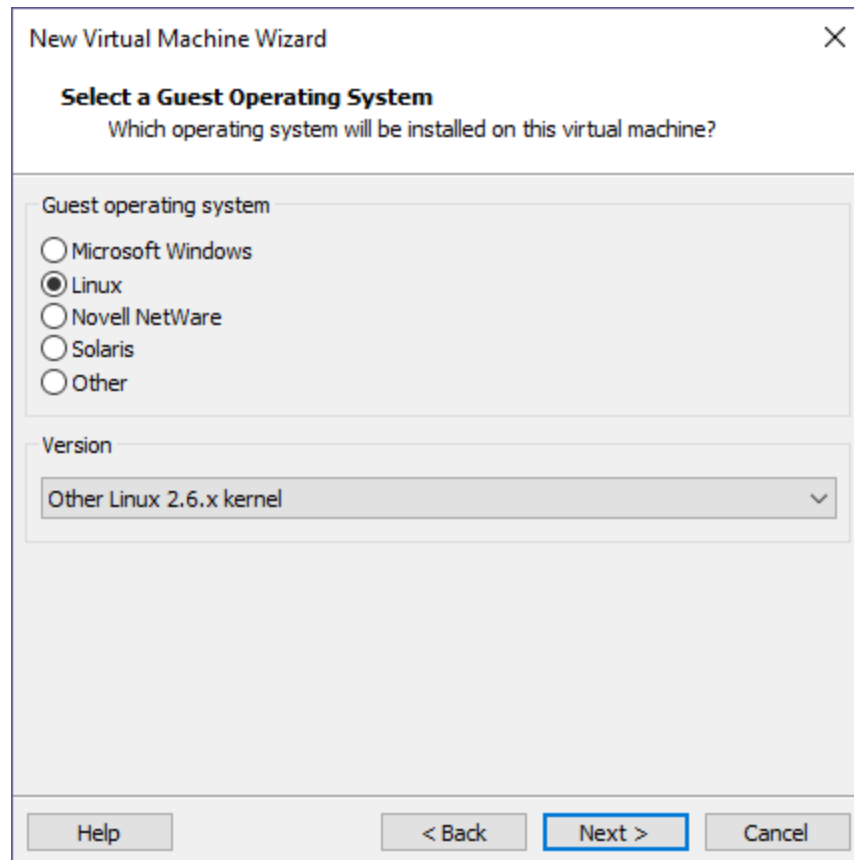
Once downloaded, we just need to go back to VMware Workstation Player and create a new virtual machine:



...select our new ISO image...



...select the OS (in this case, it's 32-bit slax, so we will chose "Other Linux 2.6.x kernel)...



...give it a name...

New Virtual Machine Wizard

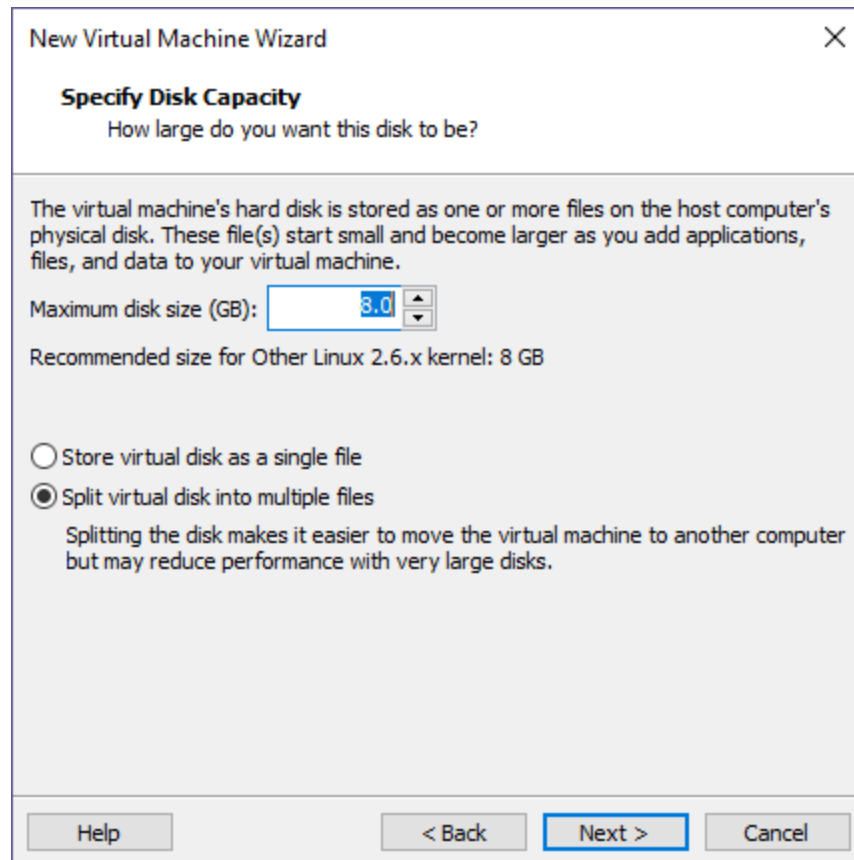
Name the Virtual Machine
What name would you like to use for this virtual machine?

Virtual machine name:
De-ICE 1.100

Location:
C:\Users\tjwil\Documents\Virtual Machines\De-ICE 1.100 Browse...

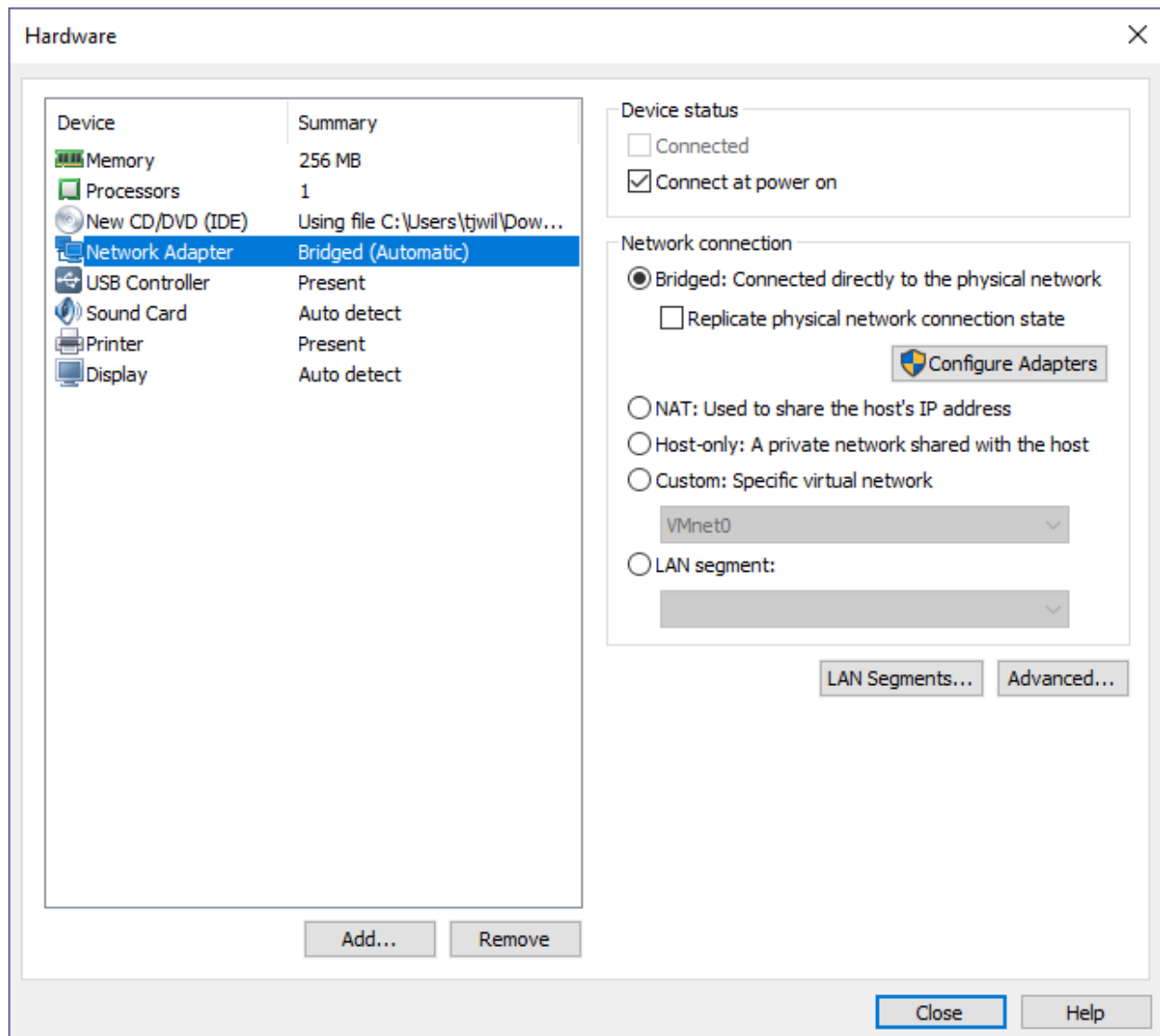
< Back Next > Cancel

...choose default settings...

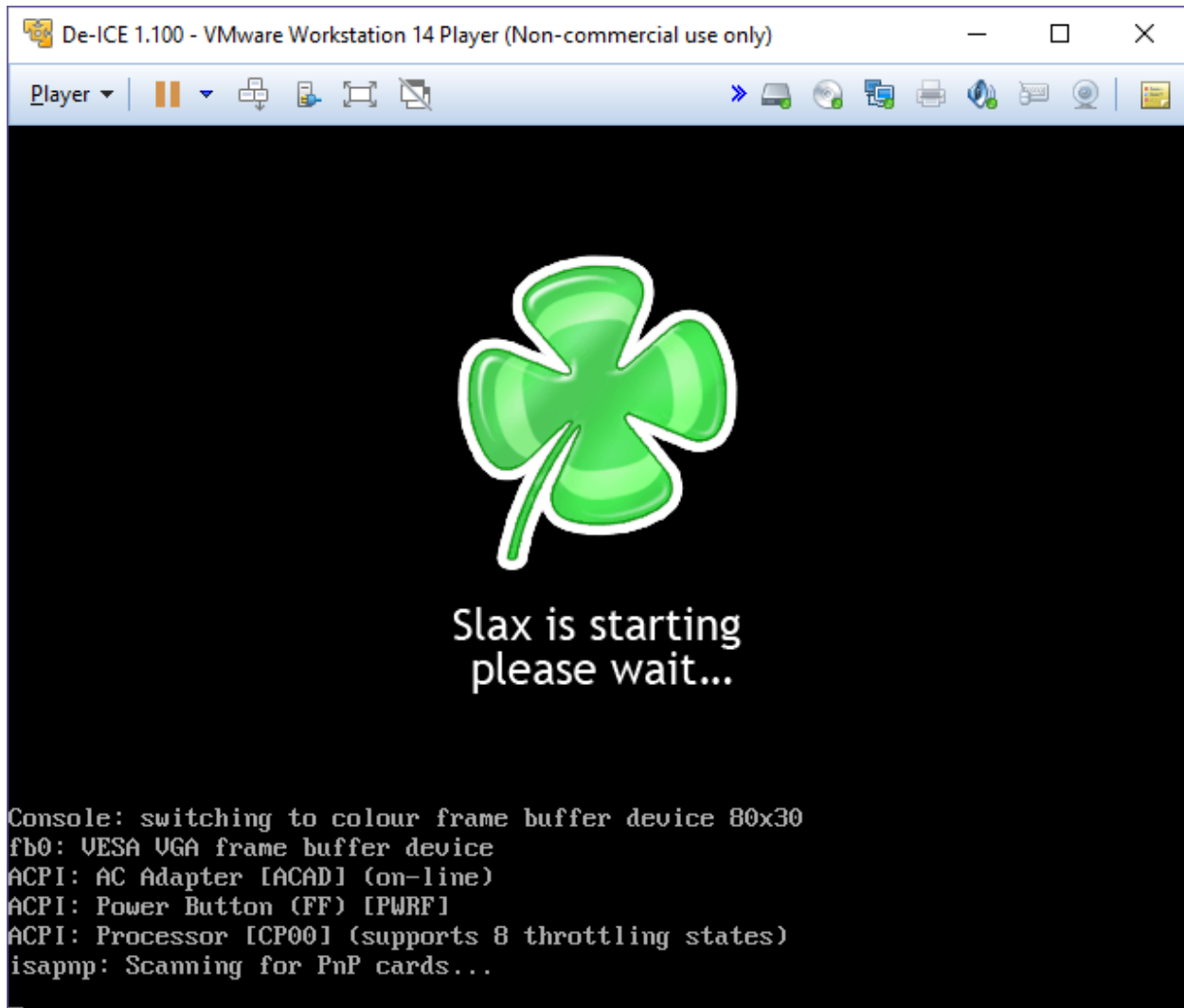


The screenshot shows a Windows-style dialog box titled "New Virtual Machine Wizard" with a close button (X) in the top right corner. The main heading is "Specify Disk Capacity" with the subtitle "How large do you want this disk to be?". Below this, a paragraph explains: "The virtual machine's hard disk is stored as one or more files on the host computer's physical disk. These file(s) start small and become larger as you add applications, files, and data to your virtual machine." A text input field for "Maximum disk size (GB):" contains the value "8.0" and has a spinner control to its right. Below the input field, it says "Recommended size for Other Linux 2.6.x kernel: 8 GB". There are two radio button options: "Store virtual disk as a single file" (which is unselected) and "Split virtual disk into multiple files" (which is selected). A note below the selected option states: "Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks." At the bottom of the dialog, there are four buttons: "Help", "< Back", "Next >" (which is highlighted with a blue border), and "Cancel".

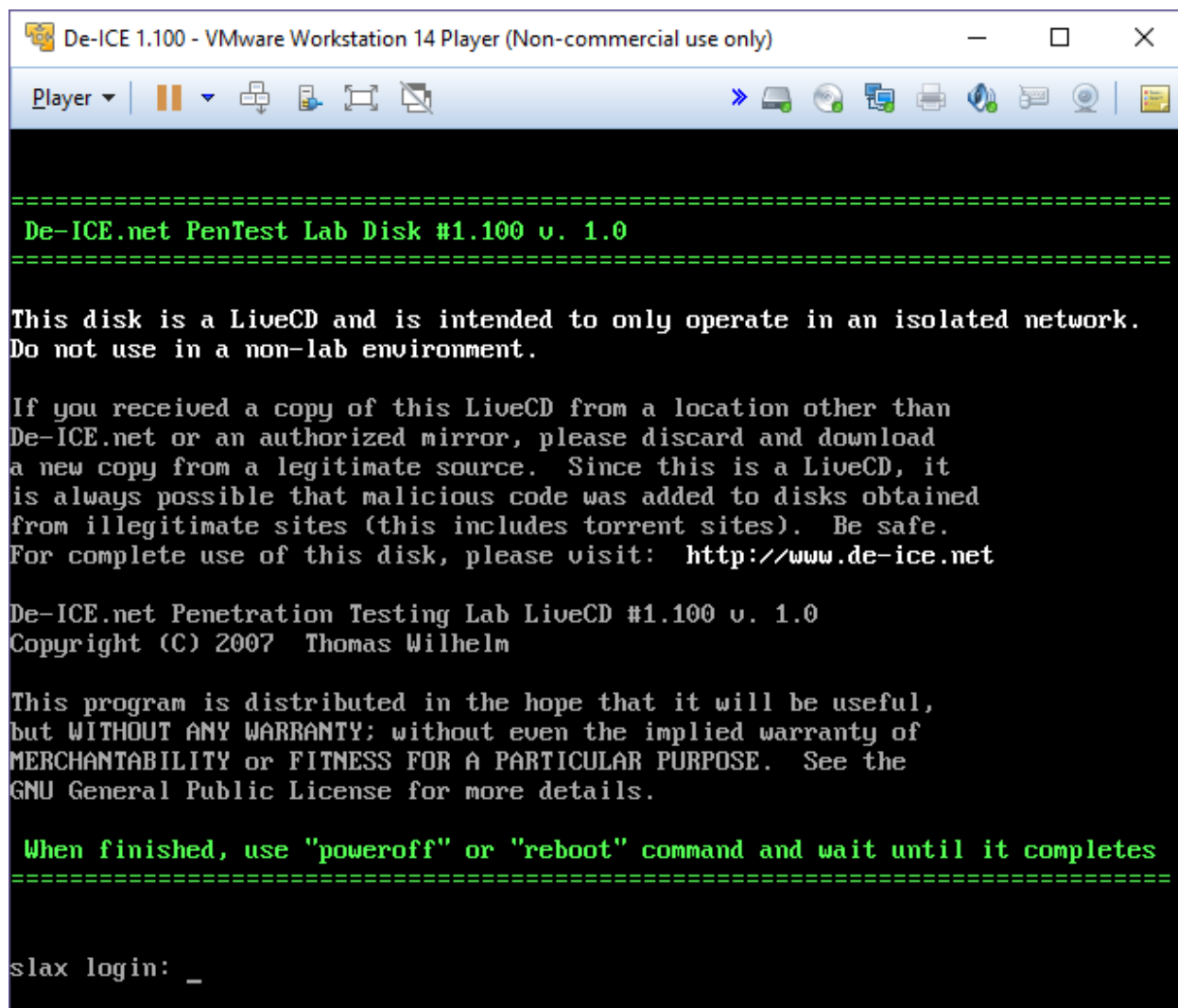
...customize the hardware to set the network adapter to “Bridged”...



...click "Finish" and the "Play virtual machine." Once started, we should see the bootup screen, similar to the one below:



The De-ICE images have pre-configured IP addresses, and the De-ICE address is 192.168.1.100. Once the system has started up we will be presented with the following login screen:



```
=====
De-ICE.net PenTest Lab Disk #1.100 v. 1.0
=====

This disk is a LiveCD and is intended to only operate in an isolated network.
Do not use in a non-lab environment.

If you received a copy of this LiveCD from a location other than
De-ICE.net or an authorized mirror, please discard and download
a new copy from a legitimate source. Since this is a LiveCD, it
is always possible that malicious code was added to disks obtained
from illegitimate sites (this includes torrent sites). Be safe.
For complete use of this disk, please visit: http://www.de-ice.net

De-ICE.net Penetration Testing Lab LiveCD #1.100 v. 1.0
Copyright (C) 2007 Thomas Wilhelm

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

When finished, use "poweroff" or "reboot" command and wait until it completes
=====

slax login: _
```

At this point, we can minimize this virtual image. Everything we do now against this target will be through Kali. If we return to Kali, we can check to see if we have connectivity to the De-ICE 1.100 image. A quick way to verify our connectivity is to bring up a web browser and navigate over to the target.

The built-in browser for Kali is available by clicking the top orange button on the quick selection bar (as indicated by the blue arrow). Once up, we type in the De-ICE IP address into the URL field, hit enter, and it looks like we have connectivity!



At this point, we have a fully functioning Personal Pentest Lab. We can repeat the process of adding additional targets to the lab as necessary.