# ENPM685 Homework #1 – Create Class Virtual Machines

Version 3.6 – January 16th 2022

**Estimated time to complete this assignment:** 4 hours

We are going to create 2 virtual machines (VMs) for use in this class:

- Kali Linux (attack VM)
- Ubuntu (victim VM)

**NOTE:** The assignments and exercises for this class are designed to be used on x86-based computers running VMWare. If you prefer to use a different architecture (such as an M1 MacBook) or virtualization technology you are welcome to but note that the instructor and TA will most likely not be able to assist you with any technical issues in your setup. These installation instructions should be architecture/virtualization agnostic but you are strongly encouraged to use an x86-based computer and VMWare for this class.

**NOTE:** If you have an existing penetration testing VM/computer running Kali/Parrot/something else that you want to use instead you can use. Submit a screenshot of your running penetration testing VM/computer instead of submitting the Kali VM set up in Part 2. You will still need to install Nessus. Also be aware that the instructor and TA will most likely not be able to assist you with any technical issues with your setup. This class is developed for and tested with Kali 2021.4 as installed in Part 2 of this assignment.

## Part 1: Install VMWare Workstation or Fusion

Download and documentation available at https://terpware.umd.edu

## Part 2: Create Kali Linux VM

**Note:** The steps below were written for the 2021.4 version but should be roughly the same for all versions of Kali. These were also written for VMWare Fusion for macOS, the VM creation process for VMWare Workstation/Windows should be similar.

1. Download Kali from https://www.kali.org
2. In VMWare create a new VM by going to the **File** menu and clicking **New…**
3. Drag the Kali ISO file to the “**Install from disc or image**” portion of the screen.
4. Click **Continue**
5. Select the Operating System, I selected “**Debian 10.x 64-bit**” for the Version since Kali is based off of Debian.
6. Make a boot firmware selection, I used “**Legacy BIOS**”
7. Click **Continue**
8. Click **Customize Settings**
9. Give the VM a name, I will use “**Kali**”

10. Select the memory size, I will use **4096 MB** (4GB) for my VM. (If you have a system with 8GB of RAM you can lower this to 2GB.)
11. For the hard drive size we'll want more than 20GB, I will use **40GB**.
12. Click the large Play button in the VM window to start up the VM and begin the install.
13. Select the install method. I recommend "**Graphical Install**"
14. Follow the instructions. Most should be pretty self-explanatory.

- For the "Configure the network" / "Domain name" you can leave that **blank**.
- Don't forget the username and password. For my example I'm using "***`kts`***" and "***`password`***"
- For disk partitioning – "**Guided – use entire disk**"
- Select the **40GB SCSI disk**
- Select "**All files in one partition (recommended for new users)**"
- "**Finish partitioning and write changes to disk**" + **Continue**
- Configure the package manager – Use a network mirror – "**Yes**"
- When asked to select and install software leave all the defaults and continue on
- Unless your home/office connection needs to use a proxy that should be left blank

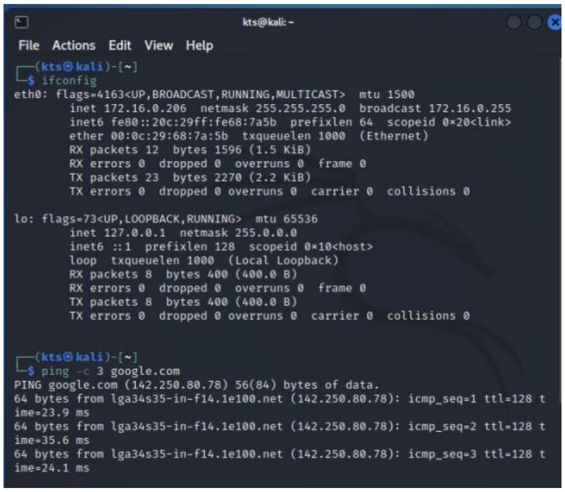
Give the installer some time to install everything. Maybe do some of the reading for this class while you wait? ☺

- "**Yes**" to install GRUB boot loader
- Select "**/dev/sda/**"

**Congrats you have built a Kali Linux VM!**

Verify you have network and connectivity:

- Login
- Open a terminal
- Run **`ifconfig`** you should see a network interface (most likely **eth0** with an IP address in an RFC1918 range (192.168.0.0/16, 172.16.0.0/12, or 10.0.0.0/8)
- Ensure you can ping another VM and/or an external site like google.com
    - Ex: **`ping google.com`**



```
┌──(kts㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.0.206  netmask 255.255.255.0  broadcast 172.16.0.255
        inet6 fe80::20c:29ff:fe68:7a5b  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:68:7a:5b  txqueuelen 1000  (Ethernet)
        RX packets 12  bytes 1596 (1.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 23  bytes 2270 (2.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 8  bytes 400 (400.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 400 (400.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


┌──(kts㉿kali)-[~]
└─$ ping -c 3 google.com
PING google.com (142.250.80.78) 56(84) bytes of data.
64 bytes from lga34s35-in-f14.1e100.net (142.250.80.78): icmp_seq=1 ttl=128 t
ime=23.9 ms
64 bytes from lga34s35-in-f14.1e100.net (142.250.80.78): icmp_seq=2 ttl=128 t
ime=35.6 ms
64 bytes from lga34s35-in-f14.1e100.net (142.250.80.78): icmp_seq=3 ttl=128 t
ime=24.1 ms
```

**Optional: Install an SSH server on the Kali VM**

I personally like the ability to connect from my host system to Kali via the command line so I'm going to install an OpenSSH server on Kali. If you want to do this go for it. Also, for a production setup I probably would not do this or I would include additional security measures like firewalling, SSH keys, and multi-factor authentication.

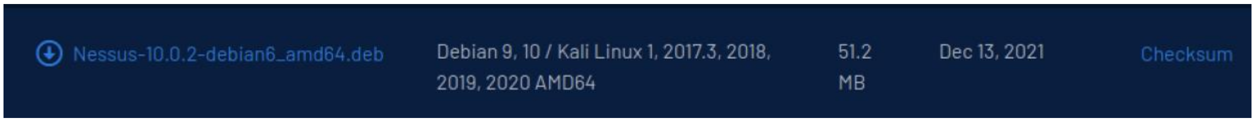
1. In Kali open up a Terminal and type **`sudo apt-get install openssh-server`**
2. Enable the SSH server to start up at boot with **`sudo systemctl enable ssh.service`**
3. Finally start the SSH server with **`sudo systemctl start ssh.service`**
6. Test everything works by SSHing into your Kali VM from your host with **`ssh`** ***`username@kali.ip`***

## Part 3: Install Nessus on your Kali VM

This will be a key tool we use during the Vulnerability Assessment week and the initial download of vulnerability checks takes some time.  To save valuable in class time you will do the initial download and setup of Nessus now.

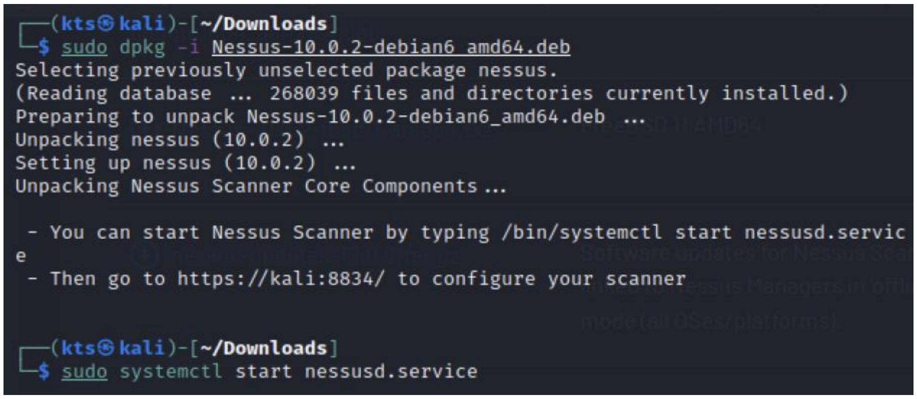
The more memory you can give to your Kali VM for Nessus the happier you'll be.  2GB should be enough.

1. Boot up your Kali VM
2. Open up **Firefox ESR** (listed as "**Web Browser**" in Kali 2020.x) and go to https://www.tenable.com/downloads/nessus
3. Scroll down and look for "**Nessus-10.0.2-debian6_amd64.deb**" with a Description of "**Debian 9,10 / Kali Linux 1, 2017.3, 2018, 2019, 2020 AMD64**" (Or if you downloaded/installed the 32bit version of Kali select "Nessus-8.9.1 -debian6_i386.deb" with a description of "Debian 9, 10 / Kali Linux 1, 2017.3 i386(32-bit)". For this example I will use the 64-bit version but the steps are the same.



| | | | | |
|---|---|---|---|---|
| Nessus-10.0.2-debian6_amd64.deb | Debian 9, 10 / Kali Linux 1, 2017.3, 2018, 2019, 2020 AMD64 | 51.2 MB | Dec 13, 2021 | Checksum |

4. Agree to the license terms by clicking "**I Agree**"
5. Save the file and let it download.
6. Open a **Terminal**
7. Type **`cd Downloads`**
8. Type **`sudo dpkg -i Nessus-10.0.2-debian6-amd64.deb`** and let Nessus install.
9. When finished type **`sudo /bin/systemctl start nessusd.service`** to start Nessus.

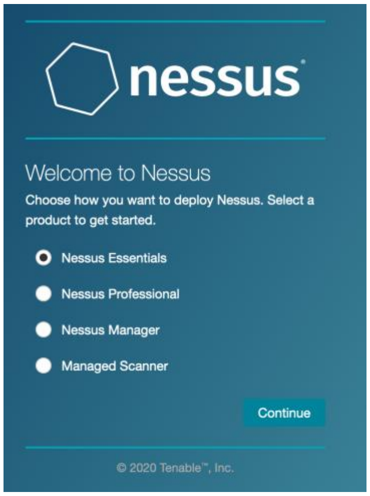


```
┌──(kts㉿kali)-[~/Downloads]
└─$ sudo dpkg -i Nessus-10.0.2-debian6_amd64.deb
Selecting previously unselected package nessus.
(Reading database ... 268039 files and directories currently installed.)
Preparing to unpack Nessus-10.0.2-debian6_amd64.deb ...
Unpacking nessus (10.0.2) ...
Setting up nessus (10.0.2) ...
Unpacking Nessus Scanner Core Components...

 - You can start Nessus Scanner by typing /bin/systemctl start nessusd.servic
e
 - Then go to https://kali:8834/ to configure your scanner

┌──(kts㉿kali)-[~/Downloads]
└─$ sudo systemctl start nessusd.service
```

10. Open a web browser (on your host system or in Kali, I'd recommend you host system for more screen real estate) and go to **`https://`*`kali.ip`*`:8834`**
11. You'll most likely get an error page that the SSL certificate is not valid. That's because it's self-signed. Go ahead and continue on. In Chrome you click "**ADVANCED**" and then "**Accept the Risk and Continue**" Other browsers will have a similar process.
12. You'll be asked to select which version of Nessus to deploy. We will use **Nessus Essentials** for this course.



Welcome to Nessus

Choose how you want to deploy Nessus. Select a product to get started.

● Nessus Essentials

○ Nessus Professional

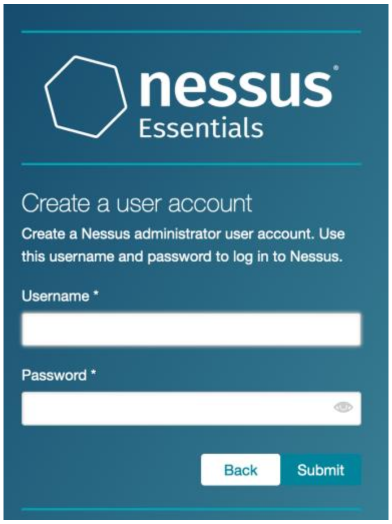
○ Nessus Manager

○ Managed Scanner

© 2020 Tenable™, Inc.

13. Fill out the information to get an email with an activation code. You'll need to supply a real email address for this, I recommend using your Terpmail account. This should be a quick process to get the email, when you have it copy and paste the code into the form. Note that the code is a one-time code. If you need to uninstall and reinstall Nessus you'll need to repeat these steps again.

14. You'll be asked to create a user account. Don't forget this information. For my example I will use:
    i. User: **`enpm685`**
    ii. Password: **`password`**



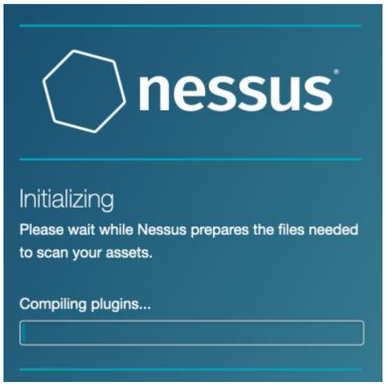
Create a user account

Create a Nessus administrator user account. Use this username and password to log in to Nessus.

Username *

Password *

15. You’ll see a message that Nessus is downloading the plugins it needs. It will take some time (around an hour or possibly more), let this complete. While you’re waiting maybe you can review some of the readings for this week. ☺

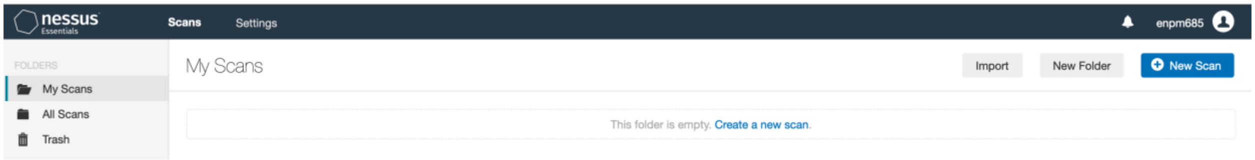


Initializing

Please wait while Nessus prepares the files needed to scan your assets.

Compiling plugins...

16. When finished Nessus will take you to the main web UI. You may get a pop up asking about starting a discovery scan you can just close that pop up.



My Scans

This folder is empty. Create a new scan.

If you want to experiment with Nessus you can, otherwise take a screenshot of the main web UI and continue on with this assignment. We’ll be working with Nessus soon enough.

## Part 4: Create Ubuntu 20.04 LTS Linux VM

**Note:** These were also written for VMWare Fusion for macOS, the VM creation process for VMWare Workstation/Windows should be similar.

1. Download the ISO file for the type of architecture your laptop is running on. Please use these specific versions. We do not want to use a newer version of Ubuntu 20.04 since the version of `sudo` in this release has a bug we want to exploit during an exercise in class.

- x86 64bit systems - https://old-releases.ubuntu.com/releases/20.04/ubuntu-20.04-live-server-amd64.iso

- M1-powered MacBooks/ARM-based systems - https://old-releases.ubuntu.com/releases/20.04/ubuntu-20.04-live-server-arm64.iso

This step will take a few minutes, probably the perfect amount of time for you to finish the readings for this week.

2.In VMWare create a new VM by going to the **File** menu and clicking **New…**
3. Drag the Ubuntu 20.04 LTS ISO file to the "**Install from disc or image**" portion of the screen.
4. Click **Continue**
5. When you see the Easy Install screen unselect it. (It doesn't work properly.)
6. Make a boot firmware selection, I used "**Legacy BIOS**"
7. Click **Continue**
8. Click **Customize Settings**
9. Give the VM a name, I will use "**`ENPM685 Ubuntu`**"
10. Select the memory size, I will use **2048 MB** (2GB) for my VM. (If you have a system with 8GB of RAM you can lower this to 1GB.)
11. Click the large Play button in the VM window to start up the VM and begin the install.
12. Most of the questions should be straightforward. Some recommendations on settings:

- When you get to the "**Guided storage configuration**" screen select "**Use an entire disk**"
- I used the following for my user information under "**Profile setup**" **Please use the same user/password.**
  - Name: **`ENPM685`**
  - Server name: **`enpm685`**
  - Username: **`enpm685`**
  - Password: **`password`**
- For the SSH Setup screen, check the box next to "Install OpenSSH server"

13. Monitor the install which will install fairly quickly. When you see “**downloading and installing security updates**” select “**Cancel update and reboot**”.

You will most likely see the VM spin for a while with “**cancelling update**” and a spinning bar. Let the VM do its job and it will reboot when it is finished. This may take 30 minutes or longer. (If you select “**View full log**” you can monitor the process, even if you cancel the update the system still attempts to update everything. This part of the install seems broken to me but most users probably do want to run the latest and greatest so this is something that hasn’t been fixed yet. ☹)

## Part 5: Configure the Ubuntu 20.04 LTS Linux VM

Now that you have the base VM we need to configure it. You can do this by downloading and running a script to configure the system to meet our needs.

1. Type the following command to download the install script to your VM: **`wget --no-check-certificate https://terpconnect.umd.edu/~kts/enpm685/install.sh`**

2. Make the script executable with **`chmod +x install.sh`**

3. Run the script with **`sudo ./install.sh`**

4. When asked if non-superusers should be able to capture packets with Wireshark select **No**

5. For configuring **snort**, select **`ens33`** as the interface. You may get an error message, if so enter **`ens33`** again and it will continue on.

6. For the network range setting for **snort** you can leave the default or enter the range of your VMWare network – if doesn’t really matter we will be using snort only to review existing packet captures. (I will use **`172.16.0.0/24`**, the range of my VMWare network.)

7. Review the Splunk License Agreement and select **`y`** to agree to it.

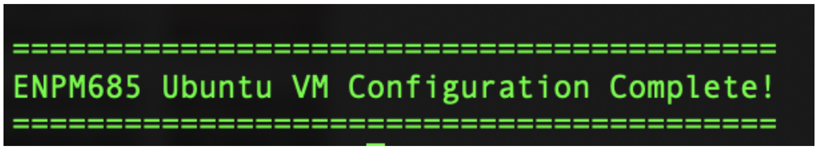
8. Enter a user name, I will use **`admin`**

9. Enter a password, I will use **`password`**

10. During the **Zeek** install you will be asked a few questions, for the **Postfix Configuration** select **`Local only`**

11. Leave system mail name as what it is defaulted to (in my case "**`enpm685`**")

12. When you see the message



```
==============================================
ENPM685 Ubuntu VM Configuration Complete!
==============================================
```

You will know that your Ubuntu VM has successfully been built.

## Part 6: Verify Connectivity and Take Screenshots

1. If they are not already running start up both VMs – Kali Linux and the Ubuntu VM.

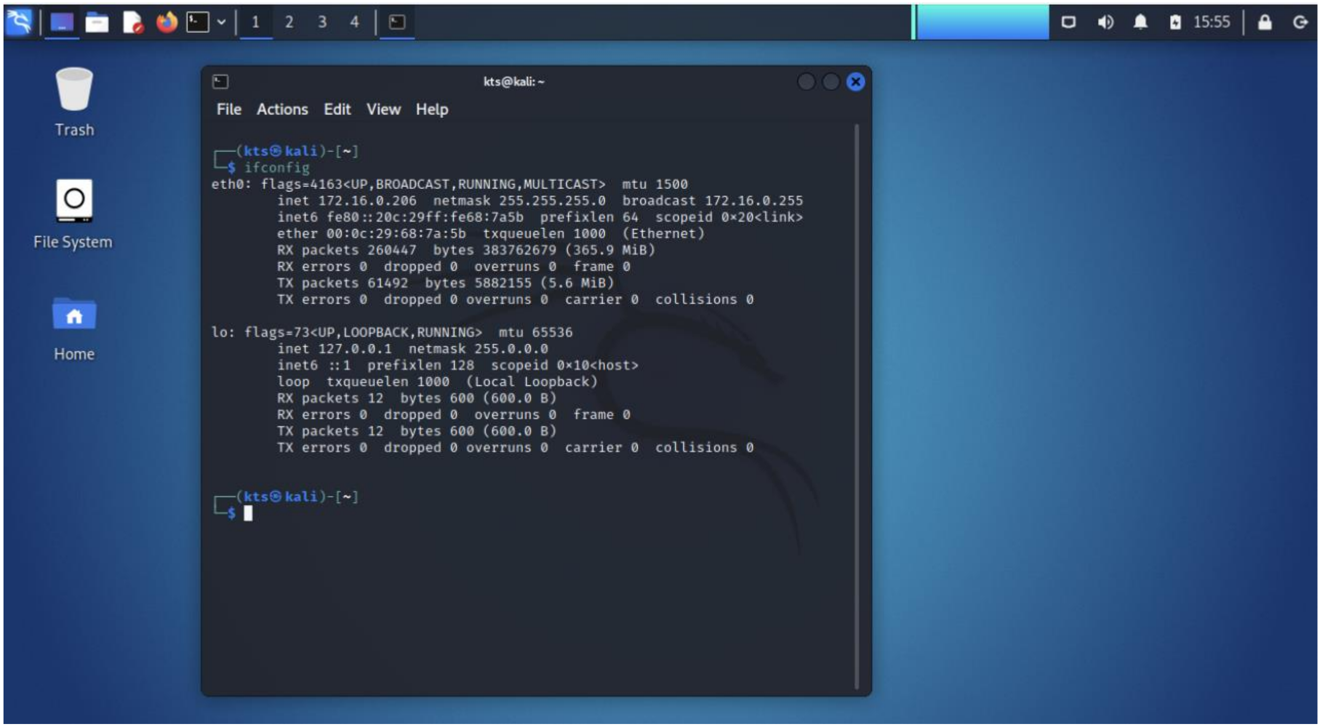
2. Once running login both VMs with the following credentials:

| | Kali | Ubuntu |
|---|---|---|
| Username | (***whatever you set***) | **enpm685** |
| Password | (***whatever you set***) | **password** |

3. Ensure each VM can ping each other and an external site (ex google.com)

4. Take a screenshot of each one of the VMs showing the output of the **`ifconfig/ip a`** commands. Your screenshots should look like:

**Kali**

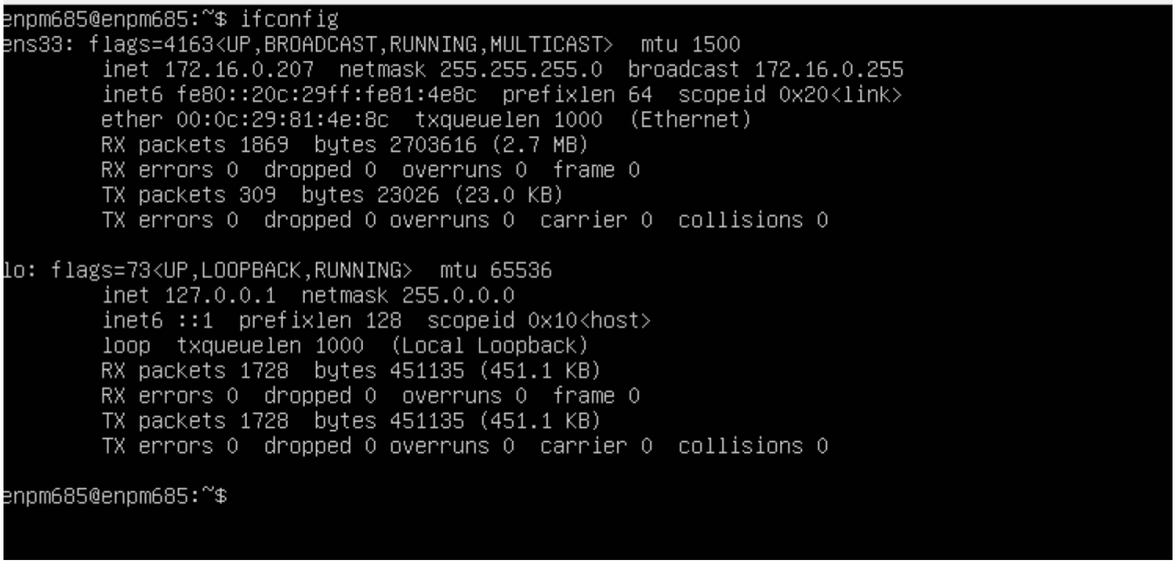


```
┌──(kts㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.0.206  netmask 255.255.255.0  broadcast 172.16.0.255
        inet6 fe80::20c:29ff:fe68:7a5b  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:68:7a:5b  txqueuelen 1000  (Ethernet)
        RX packets 260447  bytes 383762679 (365.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 61492  bytes 5882155 (5.6 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 12  bytes 600 (600.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 600 (600.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0


┌──(kts㉿kali)-[~]
└─$ 
```

**Ubuntu**

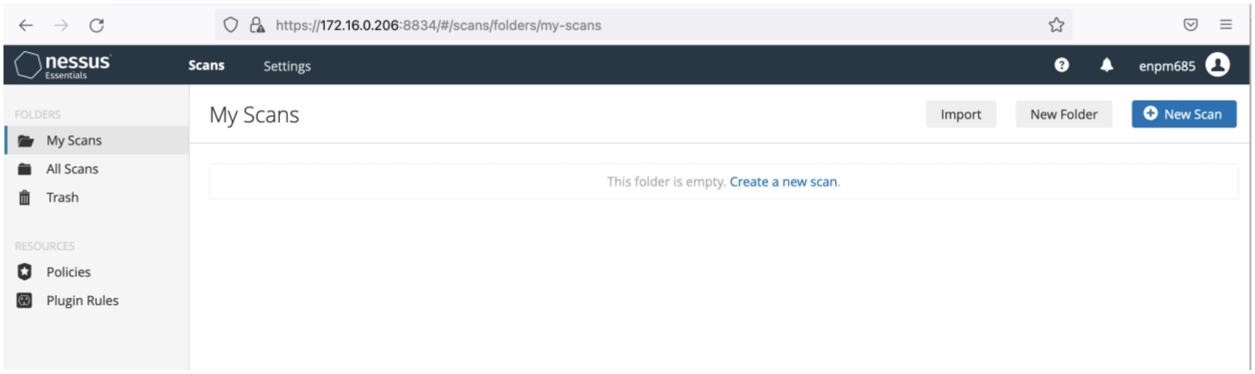


```
enpm685@enpm685:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.0.207  netmask 255.255.255.0  broadcast 172.16.0.255
        inet6 fe80::20c:29ff:fe81:4e8c  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:81:4e:8c  txqueuelen 1000  (Ethernet)
        RX packets 1869  bytes 2703616 (2.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 309  bytes 23026 (23.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 1728  bytes 451135 (451.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1728  bytes 451135 (451.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enpm685@enpm685:~$
```

5. Take a screenshot of the Nessus web interface running on your Kali VM.  Example:



This folder is empty. Create a new scan.

6. Put these screenshots into a document that has your name, UID number, course/section information and submit that document (Word .docx or PDF) in ELMS. (See the **Deliverable** section for full details)

## Part 7: Clean up

Once you have confirmed your VMs work you can delete the following to save diskspace:

- The Kali .iso file
- The Ubuntu .iso file

## Deliverable:

Submit a document (PDF or Word .docx) containing

- Your name
- UID number (ex **103946946**)
- Course and section information (Ex: **ENPM605 0101**)
- The screenshot of your Kali VM
- The screenshot of your Ubuntu VM
- The screenshot of the web interface of Nessus running on your Kali VM

These screenshots will demonstrate you have successfully built both VMs used for this course.