CO331: Network and Web Security

Setting up VirtualBox and Kali[*]

January 25 2021

This guide explains how to set up your own virtual security lab. A virtual security lab consists of a network of virtualised environments in which you can install operating systems and safely host vulnerable web applications, perform penetration tests and try out exploits, analyse malicious software without the risk of it compromising computers outside your lab, and much more. One of the environments in your lab will contain an installation of *Kali Linux*, a Linux distribution focused on providing high-quality security tools that are vital for any security lab. You'll also be able to save the state of any of the lab's virtualised environments and roll them back to these states at any time, in the common case where you need to revert part of the lab to a known safe state.

When following this guide, text typeset like `this` should be entered on the command line.

1 Installing virtualisation software

The first step is to install some virtualisation software; this provides the sandbox in which other operating systems can run. Popular choices include VMware Workstation (for Windows and Linux), VMware Fusion (for OS X), and VirtualBox (for all three). You'll install the virtualisation software in your main operating system (i.e. the one with direct access to and control of your hardware); this is known as the *host OS*. This will allow you to run an operating system inside a virtualised environment (a *virtual machine*, or *VM*); this one is known as the *guest OS*.

For the labs in this course we use VirtualBox, because it's free and works on any operating system you're likely to be running. You can download the VirtualBox installer (or package) for your host OS from https://www.virtualbox.org/wiki/Downloads, and then run the installer (or install the package using your package manager, if you downloaded a package for your Linux distribution). If you downloaded the "all distributions" self-extracting installer for Linux, you'll need to make the installer executable (e.g. `chmod +x`) as root. Follow the instructions to install VirtualBox — use the default settings suggested by the installer, and if you're given the option of installing network drivers, install them all.

2 Setting up a new virtual machine

After you have some virtualisation software installed, you can start creating VMs. VirtualBox creates VMs inside your home directory. As you use a VM, its *virtual disk* — a regular file (or several files) visible to the host OS that constitute the raw disk visible to the guest OS — will begin to grow.

We'll now create a VM in which to run Kali Linux as a guest OS; since Kali is derived from the Debian Linux distribution, for this VM we'll instruct VirtualBox to virtualise hardware that is known to be supported by Debian.

1. Open **VirtualBox Manager**.
2. Click **New**.

[*] Evolved from the original guide by Chris Novakovic c.novakovic@imperial.ac.uk. Some links and commands may have changed slightly over time.

3. Give your new VM a name (e.g. `kali-vm`). Choose **Linux** as the type, and **Debian (64 bit)** as the version. Click **Next**.
4. Decide how much memory to allocate to `kali-vm` while it is powered on. 768MB is the minimum amount you'll need to run Kali according to the developers, although you can allocate more if you have enough memory. Most machines in the main lab have 16GB of memory, so if you're following this guide in the lab you can usually safely allocate 4GB of memory to `kali-vm`. Click **Next**.
5. Make sure **Create a virtual hard drive now** is checked, and click **Create**.
6. Choose to create a virtual disk in **VDI (VirtualBox Disk Image)** format, and click **Next**.
7. Make sure you create a disk with **dynamically allocated** space (this means the virtual disk space won't all be allocated in your host OS now: the size of the virtual disk will grow as you write more data to disk in the guest OS), and click **Next**.
8. Decide how much space to allocate to `kali-vm`'s virtual disk. You'll need a minimum of 15GB, although you can allocate more if you have more free disk space available. The VMs you'll use in future tutorials need around 4GB of disk space in total, so if you're on a lab machine, bear this in mind when deciding how much space to allocate to `kali-vm` on your external drive. Click **Create**.
9. VirtualBox will create a new blank VM named `kali-vm` with the settings you gave. Right-click on `kali-vm` in the left-hand list and choose **Settings**.
10. Choose **System**, go to the **Processor** tab, and drag the **Processor(s)** slider to **2 CPUs** if the area beneath it is green; this will direct more of the real CPU's resources to `kali-vm`, so Kali will run more smoothly. Check the **Enable PAE/NX** box to expose these capabilities of the CPU to the VM.
11. Choose **Display**, go to the **Video** tab, and drag the **Video Memory** slider to **32MB** if the area beneath it is green.
12. Choose **Storage**, in the *Storage Tree* list click **Controller: SATA**, and check **Use Host I/O Cache** to greatly improve disk performance in the guest OS. Click **OK**.

3 Installing Kali Linux as a guest OS

Now it's time to install a guest OS in the VM you just created. You can install any OS of your choosing (provided it's supported by your virtualisation software — OS X generally isn't, at least not officially), but in this guide we'll install Kali Linux, a Linux distribution with a focus on penetration testing. It contains many tools you'll find useful in a fully-functional security lab.

1. Download a copy of the current *Kali Linux 64-bit ISO* installer from `https://www.kali.org/downloads/`.
2. That ISO file is a DVD image, and VirtualBox can *mount* it so that it appears as a DVD inserted into a virtual DVD drive attached to a VM. In VirtualBox Manager, right-click on `kali-vm` in the left-hand list and choose **Settings**.
3. Choose **Storage**, and click on the first child item under **Controller: IDE** in the **Storage Tree**. It'll probably be named **Empty**.
4. Click 💿▾, then **Choose/Create a Virtual Optical Disk…**.
5. Locate the ISO file, then press **Open**.
6. You should see that the child item you clicked under **Controller: IDE** in the **Storage Tree** is now named after the ISO file you downloaded. Click **OK**.
7. Click **Start**. A new window will appear, and the `kali-vm` VM will start booting. The VM prioritises the virtual DVD drive over the virtual disk when booting, so the Kali boot menu will appear.
8. Install Kali by following the steps in the *Kali Linux Installation Procedure* section of the *Kali Linux Hard Disk Install* guide at `https://www.kali.org/docs/installation/hard-disk-install/`, following the screenshots where necessary. We recommend choosing `kali-vm` as the hostname of the system for

consistency. If a domain name is prompted, leave it empty. You will be asked for a root password — choose one, and remember it. If asked about HTTP proxy information, leave it empty. The installer sometimes appears to be stuck on the *Configure the package manager* screen with no status updates for quite some time, but it's working in the background. When prompted to select the disk to install the bootloader onto, choose the only disk in the list.

9. When the installer reboots the VM, VirtualBox will *unmount* the ISO file you mounted in steps 2–5. After rebooting, `kali-vm` will boot into its new guest OS. You can suspend, restart or shut down Kali by clicking the power button in the top-right corner.

Now it's time to update `kali-vm`, to obtain the latest packages and updated exploits for security tools that are distributed with Kali (you may want to do this periodically to ensure you're up to date).

1. At the `kali-vm` login screen, log in with the username `root` and the password you chose when you installed Kali.
2. Open a terminal and append to the file `/etc/apt/sources.list` the following line:

```
deb http://http.kali.org/kali kali-rolling main contrib non-free
```

3. Run the command `apt-get update && apt-get upgrade -y`
4. Some updates may display a dialog box prompting you for a configuration change — at this stage you can select **Yes**. For *wireshark-common*, when asked **Should non-superusers be able to capture packets?**, select **Yes**.
5. Reboot `kali-vm` if necessary.

4 Installing the VirtualBox Guest Additions in the guest OS

Your `kali-vm` VM should be in a working state, but you might have noticed that using the guest OS isn't a very enjoyable experience: the screen resolution is low (and doesn't change when you resize the `kali-vm` window), windows tear when you drag them, and everything just feels rather slow. This is because Kali doesn't currently know how to make the most of the virtualised hardware VirtualBox is presenting to it. The *VirtualBox Guest Additions* are a collection of device drivers and programs designed to be installed in guest OSes to provide tighter integration with the hardware and host OS to improve the guest OS's performance (they also allow you to do things such as share clipboard data and files between the host and guest OSes for greater convenience, although these features are a potential security risk and should be enabled with caution, especially on VMs that you plan to infect with malware). To find out more about what the Guest Additions do and how they work, see Chapter 4 of the VirtualBox manual.

1. From the VirtualBox VM menu, click on **Devices** and **Insert Guest Additions CD image...**
2. Right click on the CD icon on the `kali-vm` desktop and select **Mount Volume**.
3. From a terminal, run the Guest Additions installer: `sh /media/cdrom0/VBoxLinuxAdditions.run`.
4. Restart `kali-vm` for the changes to take effect. You should notice the screen resolution increasing when you resize the `kali-vm` VirtualBox window, the mouse cursor disappearing on the `kali-vm` desktop when you move it off the edge of the VirtualBox window, and that the interface is more responsive than it was before.

5 Taking a snapshot of the VM's state

You now have a (mostly)working Kali Linux installation in your VM. This is a good time to take a *snapshot* of `kali-vm`'s current state. Think of snapshotting as hibernation for a VM, except you can restore it to a condition it was previously in whenever you want — this is a very useful feature that will allow you to "roll back" `kali-vm` to a known working state if something goes wrong. A snapshot will also store the VM's settings, and restoring a snapshot will also revert the VM's settings to those that were set when the snapshot was taken.

1. Make sure `kali-vm` is stopped (shut down Kali if necessary).
2. In VirtualBox Manager, click on `kali-vm` in the left-hand list and click **Snapshots** in the top-right corner.
3. Take a snapshot of `kali-vm`'s current state by clicking 📷.
4. Give this snapshot a name (e.g. **Installed Kali**) and optionally a description, then click **OK**.
5. You should see that `kali-vm`'s current state becomes a child of the snapshot you just created. Return to the *Details* panel by clicking **Details** in the top-right corner.

Should you ever need to revert `kali-vm`'s state to this snapshot in future:

1. In VirtualBox Manager, click on `kali-vm` in the left-hand list and make sure it isn't currently running (shut down Kali if necessary).
2. Click **Snapshots** in the top-right corner.
3. Select the snapshot you want to restore, then click 🔄 to restore the snapshot.
4. You will be asked if you want to take a snapshot of `kali-vm`'s current state. If you choose not to do this, any changes you've made to the VM since you last took a snapshot will be lost. Make your choice and click **Restore**.
5. After the snapshot is restored, return to the *Details* panel by clicking **Details** in the top-right corner.