METASPLOITABLE ON VIRTUALBOX WITH CLI

Having the ability to target a machine without going to jail is a good thing. There are lots of these types of setups available, but one of the more well-known is the 'metasploitable' setup (see here). It is a great way to get to know your way around *metasploit* and practice some basic methods. More importantly, it is made to be run as a virtual machine. So what I am going to go through now is how to setup a 'attack' machine (kali) and a 'target' machine using virtualbox and the command line (why the command line? well, just because really).

First we need to get *virtualbox*, if I assume you are using ubuntu or debian, it would be:

```
wget -q http://download.virtualbox.org/virtualbox/debian/oracle_vbox.asc -O- | sudo apt-key add -
sudo apt-get update
sudo apt-get install virtualbox-4.2
```

Then you want to get the extensions setup for the best functionality:

```
wget http://download.virtualbox.org/virtualbox/4.2.12/Oracle_VM_VirtualBox_Extension_Pack-4.2.12-84980.vbox-extpack
```

Now we will install it. This is the first use of the *VBoxManage* command, but not the last. This is the main command we will be using as it is able to do everything the GUI can:

```
VBoxManage extpack install --replace ./Oracle_VM_VirtualBox_Extension_Pack-4.1.6-74713.vbox-extpack
```

To start we will register our new virtual machine and setup the RAM and OS type. We also turn on *pae* to help with systems that expect newer processor features. We will also tell it to boot from the DVD:

```
VBoxManage createvm --name "kali" --register
VBoxManage modifyvm "kali" --memory 512 --acpi on --boot1 dvd
VBoxManage modifyvm "kali" --ostype Debian
VBoxManage modifyvm "kali" --pae on
```

Now we want to setup the networking. Virtualbox has a few options, but for these machines I want them to communicate with one another and the host but nothing else. So we do this:

```
VBoxManage hostonlyif create
VBoxManage hostonlyif ipconfig vboxnet0 -ip 10.10.10.1 -netmask 255.255.255.0
VBoxManage modifyvm "kali" --nic1 hostonly
VBoxManage modifyvm "kali" --hostonlyadapter1 vboxnet0
```

The first 2 commands above are a once-off, they setup the virtual interface the host will use to communicate with the guests. Now lets setup the storage for our virtual machine:

```
VBoxManage createhd --filename /virtuals/kali/kali.vdi --size 15000
VBoxManage storagectl "kali" --name "IDE Controller" --add ide
VBoxManage storageattach "kali" --storagectl "IDE Controller" --port 0 --device 0 --type hdd --medium /virtuals/kali/kali.vdi
VBoxManage storageattach "kali" --storagectl "IDE Controller" --port 1 --device 0 --type dvddrive --medium /virtuals/kali-linux-1.0.2-i386.iso
```

Now when you start your virtual machine:

```
VBoxHeadless --startvm "kali" &
```

It will boot from the DVD and will enable RDP access to itself on port 3389. Once you have finished the install and want to eject the DVD (once the virtual machine is powered down), you can use this:

```
VBoxManage modifyvm "kali" --dvd none
```

And of course to shutdown your virtual machine, you can use:

```
VBoxManage controlvm "kali" poweroff
```

In your virtual machine you want to setup the networking so that it is using the same setup as your host-only virtual interface, and since '*kali*' is a debian derivative, we need to edit the */etc/network/interfaces* file and add this:

```
auto eth0
iface eth0 inet static
address 10.10.10.10
netmask 255.255.255.0
gateway 10.10.10.1
```

Now when the virtual machine restarts or you rerun the networking scripts, your *eth0* will be setup to communicate with the host and any other host-only virtual machines. So this gives us our '*attack*' machine, lets get our '*target*' machine setup. When you download '*metasploitable*' (here), you get a zip file that contains a *vmware* virtual machine. Now using the GUI to use/convert this file is easy, but we are using the command line. So we will use the *qemu* tools to convert it into a format virtualbox can convert:

```
apt-get install qemu
qemu-img convert ./Metasploitable2-Linux/Metasploitable.vmdk ./msfable.bin
VBoxManage convertdd ./msfable.bin ./msfable/msfable.vdi
```

Now we are going to use the same steps as above to create our new '*target*' virtual machine:

```
VBoxManage createvm --name "msfable" --register
VBoxManage modifyvm "msfable" --memory 512 --acpi on
VBoxManage modifyvm "msfable" --pae on
VBoxManage modifyvm "msfable" --nic1 hostonly
VBoxManage modifyvm "msfable" --hostonlyadapter1 vboxnet0
VBoxManage storagectl "msfable" --name "IDE Controller" --add ide
VBoxManage storageattach "msfable" --storagectl "IDE Controller" --port 0 --device 0 --type hdd --medium /virtuals/
msfable/msfable.vdi
```

And the startup and shutdown uses the sames commands as above:

```
VBoxHeadless --startvm "msfable" &
VBoxManage controlvm "msfable" poweroff
```

One new thing though, if you start the machine up now, you will get an error because the first virtual machine is already listening on port 3389. So we need to change the port our second virtual machine listens on:

```
VBoxManage modifyvm "msfable" --vrdeport 3390
```

Now when we startup we can access the 'target' virtual machine's RDP access on port 3390. Once the '*metasploitable*' virtual machine starts up, we will need to login as *msfadmin/msfadmin* and setup the networking. Since '*metasploitable*' is also a debian derivative, we can edit */etc/network/interfaces*:

```
auto eth0
iface eth0 inet static
address 10.10.10.20
netmask 255.255.255.0
gateway 10.10.10.1
```

After all is done, we have two virtual machines, each accessible via RDP (for remote RDP access, ssh onto the host and port forward the RDP ports) - one '*attack*' virtual machine on 10.10.10.10 using '*kali*' and one '*target*' virtual machine using '*metasploitable*' on 10.10.10.20. Give it a go and have fun.