Overview

VBoxManage is the command-line interface to VirtualBox. Using VBoxManage one can completely control VirtualBox. VirtualBox can be controlled from the command line of the host operating system. VBoxManage supports all the features that can be access by graphical user interface. Along with that VBoxManage also supports some additional features that cannot be accessed by graphical user interface. VBoxManage also exposes all the features of the virtualization engine, even those that cannot be accessed from the GUI.

Snort is most famous for being a full-fledged open-source network based intrusion detection system (NIDS). Snort is also a feature-rich packet sniffer and a useful packet logger. In addition to these, few central features of Snort are, (i) Snort supports sending real-time alerts when an intrusion event is detected (ii) Snort can even be used as an inline "intrusion prevention system" that enables to receive alerts in real time and in several different mediums, rather than having to continuously sit at a desk monitoring the Snort system 24 hours a day. To help one better understand the different features and capabilities of Snort, let's take a look at it by analogy. Snort is like a vacuum that sucks up all items of a particular kind (in this case, packets).Snort allows one to do different things to them once they are captured.

Snort can be used to watch the items as they get sucked up to see what is captured (also known as packet sniffer). It can also be used to put the items into a container for later examination once they are captured. This is also referred as packet logger. It is also used to sort them, match the items with a list of criteria and it lets the user know when a matching item has gone through also known as Network Intrusion Detection System (NIDS). These features allow for various types of useful security analysis to be performed, including closer examination of the contents of potential attacks (from the NIDS), live traffic sampling of ongoing security evens (from the packet sniffer), and historical data on past network events (from the packet logger).

The **Metasploitable** is the Linux virtual machine. It is intentionally vulnerable version of Ubuntu Linux Designed for testing security tools and demonstrating common vulnerabilities. This virtual machine is compatible with VMWare, Virtual Box, and other common virtualization platforms. Metasploitable mainly focuses on vulnerabilities at the operating system and network services layer. Version 2 of Metasploitable is available on source forge link which is given below. Metasploitable 2 ships with even more vulnerabilities than the original image. By default, Metasploitable's network interfaces are bound to the NAT and Host-only network adapters, and the image should never be exposed to a hostile network.

(Note: A manual on installing Metasploitable 2 is available in this document

Purpose: The main purpose is to detect network attacks using existing adaptive IDS technology (snort) and respond them.

Setup VMBox on Development server

Requirements to set up VMBox on Development Server.

- Download Ubuntu ISO
- Configure Ubuntu ISO on VMBox (one can find iso on development server-soecsdev.nyit.edu as well).

Following are the steps for setting up Ubuntu Linux on VMBox:

Step 1:

- Download ubuntu-14.04.3-server-i386.iso from link:

  https://sourceforge.net/projects/ids-snort-v1/files/ubuntu-14.04.3-server-i386.iso/download

Step 2:

- Once the Ubuntu is downloaded the next Step is to setup VMBox on development server for **testvm1**.
- Execute the following commands to setup VMBox on development server (for testvm1):

VBoxManage list runningvms

VBoxManage createvm --name "testvm1" --register

VBoxManage modifyvm "testvm1" --memory 1024 --acpi on --boot1 dvd --nic1 bridged --bridgeadapter1 em1 --ostype Ubuntu

VBoxManage showvminfo "testvm1" --details

VBoxManage modifyvm "testvm1" --vrde on --vrdeport 3390

VBoxManage createvdi --filename ~/VirtualBox\ VMs/testvm1/testvm1-disk01.vdi --size 10000

VBoxManage storagectl "testvm1" --name "IDE Controller" --add ide

VBoxManage storageattach "testvm1" --storagectl "IDE Controller" --port 0 --device 0 --type hdd --medium ~/VirtualBox\ VMs/testvm1/testvm1-disk01.vdi

VBoxManage storageattach "testvm1" --storagectl "IDE Controller" --port 1 --device 0 --type dvddrive --medium /home/ypatel20/iso/ubuntu-14.04.3-server-i386.iso

VBoxManage modifyvm "testvm1" --vrde on

VBoxHeadless --startvm "testvm1" &

VBoxManage controlvm testvm1 poweroff

Step 3:

- Once the commands for setting up VMBox for **testvm1** is executed on development server execute **testvm2**.
- Execute the following commands to setup VMBox on development server (for testvm2)

VBoxManage list runningvms

VBoxManage createvm --name "testvm2" --register

VBoxManage modifyvm "testvm2" --memory 1024 --acpi on --boot1 dvd --nic1 bridged --bridgeadapter1 em1 --ostype Ubuntu

VBoxManage modifyvm "testvm2" --vrde on --vrdeport 3389

VBoxManage createvdi --filename ~/VirtualBox\ VMs/testvm2/testvm2-disk01.vdi --size 10000

VBoxManage storagectl "testvm2" --name "IDE Controller" --add ide

VBoxManage storageattach "testvm2" --storagectl "IDE Controller" --port 0 --device 0 --type hdd --medium ~/VirtualBox\ VMs/testvm2/testvm2-disk01.vdi

VBoxManage storageattach "testvm2" --storagectl "IDE Controller" --port 1 --device 0 --type dvddrive --medium /home/ypatel20/iso/ubuntu-14.04.3-server-i386.iso

VBoxManage modifyvm "testvm2" --vrde on

VBoxHeadless --startvm "testvm2" &

VBoxManage controlvm testvm2 poweroff

When both Step 2 and Step 3 are executed successfully, it shows that VMBox configurations are installed successfully. The next step will be to install Ubuntu Linux using GUI interface tool (Tight VNC client).

TightVNC client can be downloaded from the link below:

https://sourceforge.net/projects/ids-snort-v1/files/tvnjviewer-2.7.2-bin.zip/download

Step 4:

- .Zip file is downloaded for TightVNC.
- After extracting .zip file and run tightvnc-jviewer.jar (To connect VNC Server using the TightVNC client with GUI)

VNC server is already installed on development server (soecsdev.nyit.edu - **198.242.56.249**) and to access it Tight VNC client is used. Try to connect VNC client through Tight VNC SSH connection.

TightVNC connection via ssh tunneling

Following is the example of credentials for TightVNC connection:

Remote Host: 198.242.56.249

Port: 3390 | 3389 SSH

Server: 198.242.56.249 SSH

Port: 22 SSH

User :(Ubuntu username)

Tight VNC client is providing GUI interface to setup both VMs on development server. Install Ubuntu operating system properly on both the VMs.

Information: following are the IPs of the virtual machines:

Development Server IP : 198.242.56.249

testvm1 IP : 198.242.56.122 tcp-port: 3390 password: testvm1

testvm2 IP : 198.242.56.123 tcp-port: 3389 password: testvm2

Then try *apt-get install update* command on terminal to check internet access is on both the VMs or not. If it isn't able to access internet on both the VMs then you nee to check proxy settings or DNS settings.

After accessing VMs using VNC client, the next step is to configure network of these VMs using the following the network settings given below:

Configure Network Settings

Proxy server is responsible for filtering the external internet traffic so you can change the network settings using the following steps. To allow *apt-get* command, change *apt.conf* file.

Change the network configuration as below:

Step 1:

- Setup proxy server settings on both the virtual machines as follows:

Proxy IP : 198.242.56.207

tcp-port : 80 (proxy.nyit.edu:80)

Step 2:

- After setting up the proxy server, setup network configuration for testvm1.
- For this, Changes has to be made in two things, that are as follows:

1. Make the following changes to /etc/network/interfaces

auto eth0
iface eth0 inet static
address 198.242.56.122
netmask 255.255.255.0
network 198.242.56.0
broadcast 198.242.56.255
gateway 198.242.56.1
 # dns- options are implemented by the resolvconf package, if installed*
dns-nameservers 64.35.176.53 64.35.176.65
dns-search nyit.edu

2. Make the following changes to /etc/apt/apt.conf

   Acquire::http::proxy "http://proxy.nyit.edu:80/";
   Acquire::https::proxy "https://proxy.nyit.edu:80/";
   Acquire::ftp::proxy "ftp://proxy.nyit.edu:80/";
   Acquire::socks::proxy "socks://proxy.nyit.edu:80/";

Information: These proxy settings are used to allow *apt-get* command on vm.

Step 3:

- After setting up the proxy server, setup network configuration for testvm2.
- For this, Changes has to be made in two things, that are as follows:

1. Make the following changes to /etc/network/interfaces

   auto eth0
   iface eth0 inet static
   address 198.242.56.123
   netmask 255.255.255.0
   network 198.242.56.0
   broadcast 198.242.56.255
   gateway 198.242.56.1
   # dns- options are implemented by the resolvconf package, if installed*
   dns-nameservers 64.35.176.53 64.35.176.65
   dns-search nyit.edu

2. Make the following changes to /etc/apt/apt.conf

   Acquire::http::proxy "http://proxy.nyit.edu:80/";
   Acquire::https::proxy "https://proxy.nyit.edu:80/";
   Acquire::ftp::proxy "ftp://proxy.nyit.edu:80/";
   Acquire::socks::proxy "socks://proxy.nyit.edu:80/";

- These proxy settings are used to allow apt-get command on vm.
- Next step will be to install OpenSSH to make connection simply from anywhere using ssh client.

Setup ssh-server on VMBox

- Execute the following command to setup ssh server on the virtual machine:

apt-get update

apt-get upgrade

apt-get install openssh-server

- Now, Ssh is installed and configured on both VMs
- Snort can be installed using command which given below:

Setup Snort on VMBox

Step 1:

- Download snort.iso file from the below link

https://sourceforge.net/projects/ids-snort-v1/files/snort.iso/download

Step 2:

- Mount the snort.iso file and copy files on machine

Step 3:

- Install snort by executing the following commands one by one

apt-get update

apt-get upgrade

apt-get install flex bison build-essential checkinstall libpcap-dev libdnet libdnet-dev libnet1-dev libpcre3-dev libmysqlclient-dev libnetfilter-queue-dev iptables-dev

Step 4:

- Execute the following commands for libnet configuration

tar xvfvz libdnet-1.12.tgz

cd libdnet-1.12

./configure CFLAGS=-fPIC; make

sudo checkinstall

sudo dpkg -i libdnet_1.12-1_i386.deb

sudo ln -s /usr/local/lib/libdnet.1.0.1 /usr/lib/libnet.1

cp /usr/local/lib/libdnet.1.0.1 /usr/local/lib/libdnet.so.1.0.1

LD_LIBRARY_PATH=/usr/local/lib

export LD_LIBRARY_PATH

Step 5:

- Execute the following commands for daq configuration

tar xvfz daq-2.0.6.tar.gz

cd daq-2.0.6/

./configure CFLAGS=-fPIC; make

sudo checkinstall

sudo dpkg -i daq_2.0.6-1_i386.deb

Step 6:

- Execute the following commands to configure snort

tar xvfz snort-2.9.8.0.tar.gz

cd snort-2.9.8.0/

./configure

sudo checkinstall

sudo dpkg -i snort_2.9.8.0-1_i386.deb

sudo ln -s /usr/local/bin/snort /usr/sbin/snort

sudo ldconfig –v

snort –V

sudo groupadd snort

sudo useradd snort -d /var/log/snort -s /sbin/nologin -c SNORT_IDS -g snort

sudo mkdir /var/log/snort

sudo chown snort:snort /var/log/snort/

Check whether snort is working properly or not using the following command:

snort

After successfully installing snort one has to also install community rules on machines. Link to get the rules are given below or can be directly installed from ISO.

wget https://www.snort.org/downloads/registered/snortrules-snapshot-2980.tar.gz

Install community rules

Step 7:

- Download snort rule from the official website manually:

mkdir /etc/snort

tar xvfvz snortrules-snapshot-2980.tar.gz -C /etc/snort/1

sudo mkdir /usr/local/lib/snort_dynamicrules

*sudo chown -R snort:snort /etc/snort/**

sudo mv /etc/snort/etc/ /etc/snort/*

Step 8:

- Edit the snort.conf using vim editor:

sudo vim snort.conf

ipvar HOME_NET 10.1.10.0/24

ipvar EXTERNAL_NET !$HOME_NET

Change RULE_PATH ../rules/ to RULE_PATH /etc/snort/rules/

Step 8: Disable firewall:

sudo ufw disable

Step 9:

- Run snort on eth0:

1. In packet dump mode:
   sudo snort -c /etc/snort/snort.conf -A console -i eth0 -K ascii

2. In sniffer mode:
   snort -vde
3. In sniffer-log mode:

   snort -vde -l /var/log/snort/ -K ascii

4. In test mode
   sudo snort -T -i eth0 -u snort -g snort -c /etc/snort/snort.conf

5. In analysis mode
   snort -A console -i eth0 -c /etc/snort/snort.conf -l /var/log/snort -K asci

6. In read pcap mode
   snort -r /var/www/html/IDS/upload_pcap/tcp.pcap -c /etc/snort/snort.conf -l /var/log/snort -K ascii

- After successfully configured snort on both the VMs Metasploit framework VM has been configured for penetration testing.

Setup Metasploit Framework on VMBox

Step 1:

- Download metasploit.iso file from the link below:

https://sourceforge.net/projects/ids-snort-v1/files/metasploit.iso/download

Step 2:

- Mount metasploit.iso and copy the files from the iso file into VMs directory

Step 3:

- Following are the steps for Metasploit environment setup:

sudo apt-get install build-essential libreadline-dev libssl-dev libpq5 libpq-dev libreadline5 libsqlite3-dev libpcap-dev openjdk-7-jre git-core autoconf postgresql pgadmin3 curl zlib1g-dev libxml2-dev libxslt1-dev vncviewer libyaml-dev curl zlib1g-dev ruby1.9.3

sudo apt-get install nmap

sudo gem install wirble sqlite3 bundler

su postgres

createuser msf -P -S -R –D
createdb -O msf msf

chmod +x metasploit-latest-linux-installer.run

./metasploit-latest-linux-installer.run

sudo msfupdate

sudo msfconsole

After successfully configuration of snort on both the VMs and also configuring on one VM, now configuring Metasploitable (Vulnerable Ubuntu Linux) on another VM.

Configure Metasploitable Ubuntu Linux on VMBox

Step 1:

- Download Metasploitable Ubuntu Linux 2.rar file from the given below link:

https://sourceforge.net/projects/ids-snort-v1/files/Metasploitable_Ubuntu_Linux-2.tar.gz/download

- Make vm named directory at following path: ~/VirtualBox\ VMs/

- Extract the downloaded .rar file at that same path. (~/VirtualBox\ VMs/vm/)

Step 2:

- Configure Metasploitable Ubuntu Linux on development server

vm IP: 198.242.56.121

tcp-port:3399

vm::vm (Vulnerable Linux)

VBoxManage list runningvms

VBoxManage createvm --name "vm" –register

VBoxManage modifyvm "vm" --memory 512 --acpi on --boot1 dvd --nic1 bridged --bridgeadapter1 em1 --ostype Ubuntu

VBoxManage showvminfo "vm" –details

VBoxManage modifyvm "vm" --vrde on --vrdeport 3399

VBoxManage storagectl "vm" --name "IDE Controller" --add ide
VBoxManage storageattach "vm" --storagectl "IDE Controller" --port 0 --device 0 --type hdd --medium ~/VirtualBox\ VMs/vm/vm-disk.vdi

VBoxManage modifyvm "vm" --vrde on

VBoxHeadless --startvm "vm" &

VBoxManage controlvm vm poweroff

Information: Configuration of proxy settings on this vm is done in the same way as the previous one.

- Network configuration for VM can be done by following steps:

1. Make the following changes to /etc/network/interfaces

   auto eth0
   iface eth0 inet static
   address 198.242.56.121
   netmask 255.255.255.0
   network 198.242.56.0

broadcast 198.242.56.255
gateway 198.242.56.1
 # dns- options are implemented by the resolvconf package, if installed*
dns-nameservers 64.35.176.53 64.35.176.65
dns-search nyit.edu

2. Make the following changes to /etc/apt/apt.conf

Acquire::http::proxy "http://proxy.nyit.edu:80/";
Acquire::https::proxy "https://proxy.nyit.edu:80/";
Acquire::ftp::proxy "ftp://proxy.nyit.edu:80/";
Acquire::socks::proxy "socks://proxy.nyit.edu:80/";

These proxy settings are used to allow apt-get command on VM.

REFERENCES

http://www.unixmen.com/install-snort-nids-ubuntu-15-04/

http://linton.tw/2014/08/17/Install-Snort-from-source-on-Ubuntu/

http://linton.logdown.com/posts/2014/08/16/install-snort-from-source-on-ubuntu

http://manual.snort.org/node8.html

https://www.virtualbox.org/manual/ch08.html

http://linuxpitstop.com/install-and-use-command-line-tool-vboxmanage-on-ubuntu-16-04/

http://www.howopensource.com/2011/06/how-to-use-virtualbox-in-terminal-commandline/

http://nakkaya.com/2012/08/30/create-manage-virtualBox-vms-from-the-command-line/