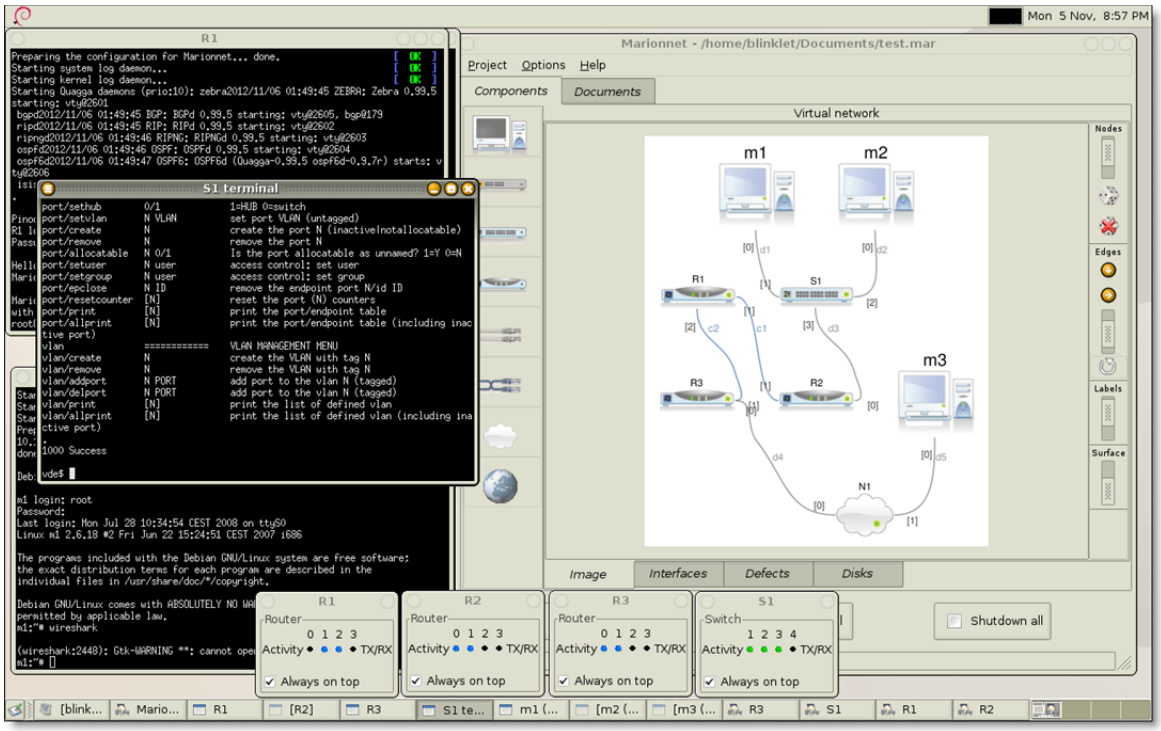
Install the Marionnet network simulator on Kali Linux Light 2017.1-vbox-i686

August 26, 2017 by Lucian Visinescu (this work is licensed under CC BY-NC 4.0)

This installation is based on Install the Marionnet network simulator on Debian Linux 6.0 by Brian Linkletter, therefore credit should also go to him. I used his text in this example and I didn't change it unless it was necessary.

The Marionnet network simulator is easy to install but there are some specific system configurations that must be changed so that all the features of Marionnet work well.



This post will describe the installation procedure for the Marionnet open-source network simulator on Kali Linux Light 2017.1-vbox-i686.

Install Marionnet

The Marionnet project offers an easy-to-use install script. Download the *marionnet_from_scratch* script and run it using the following commands:

```
$ wget http://www.marionnet.org/downloads/marionnet_from_scratch/marionnet_from_scratch
$ chmod +x marionnet_from_scratch
$ sudo ./marionnet_from_scratch
```

The *marionnet_from_scratch* script will ask for some responses to configuration questions. Enter "Y" at all the prompts. You may be asked to enter your root password at one point if you are not root when you do the installation. The script will install all the required prerequisite packages; download, compile, and install the Marionnet software; and download and install the filesystems and kernels for the Linux virtual machines used by the Marionnet network simulator.

Note: The *marionnet_from_scratch* script will download over *a gigabyte* of data, most of which is filesystems and compiled kernels for the virtual machines.

After the script completes, logout and login to ensure all changes are applied.

The Marionnet Daemon

The *Marionnet daemon* is a server that requires root privileges and provides some system-level services for Marionnet.

The *Marionnet daemon* supports networking between the virtual machines created by Marionnet and the host operating system which allows the virtual machines to run programs with a graphical user interface (GUI), such as Wireshark, and display the GUI on an X server window on the host system, making user interactions with these programs possible.

The *Marionnet daemon* also allows Marionnet virtual machines to connect to the host's Ethernet card and communicate with computers or other equipment on an external network.

There are two ways to ensure the *Marionnet daemon* will be running before you start the Marionnet GUI program:

- Start the daemon manually when you want to use Marionnet, or
- Configure your host system to launch the daemon when the host system is started.

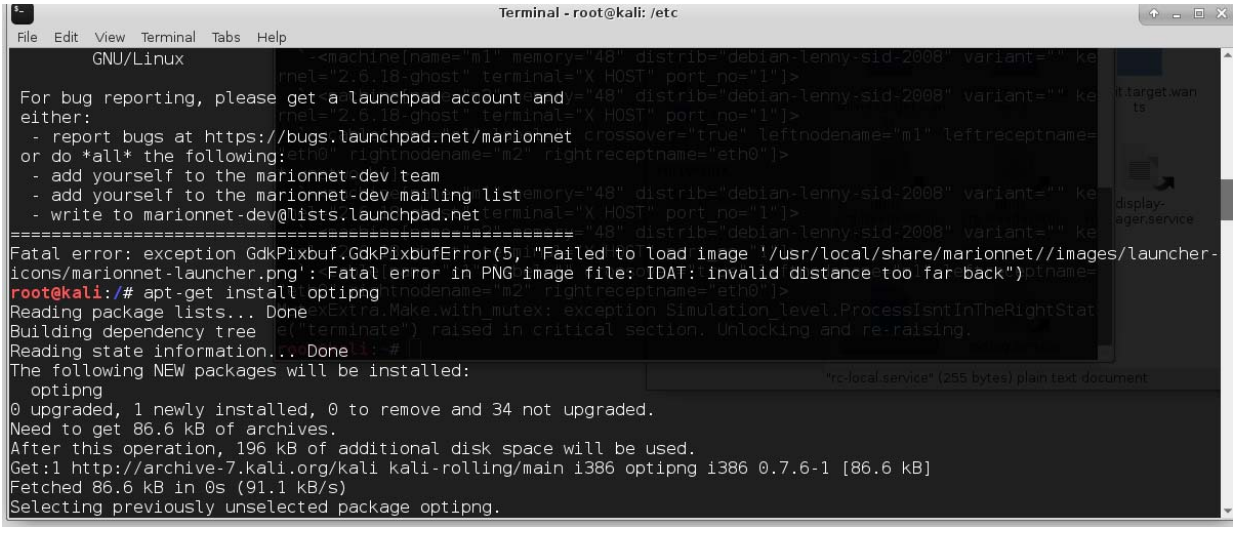
Option #1: Manually start the *Marionnet daemon*

You may choose to manually start the *Marionnet daemon* each time you want to run Marionnet.

To manually start the daemon, enter the following command either as super-user or as the *root* user. In this case, we will use the `sudo` command to run `marionnet-daemon.byte` with super-user privileges.

```
$ sudo marionnet-daemon.byte
```

Chances are that first time you start the daemon you will get some errors and warnings in your terminal window. One of the errors is

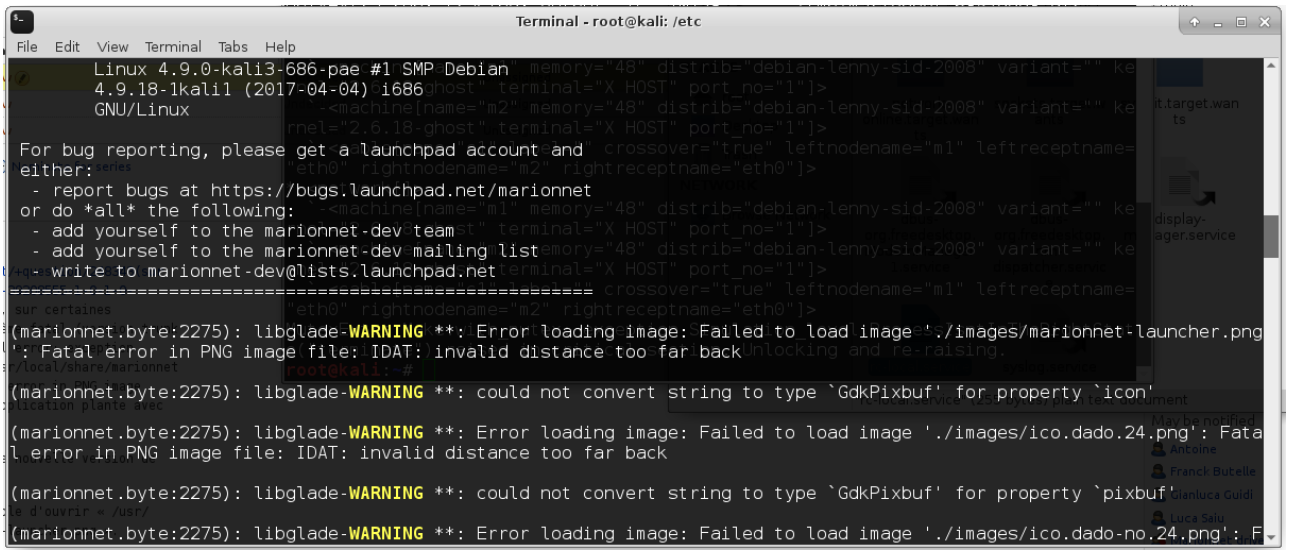


```
Fatal error: exception GdkPixbuf.GdkPixbufError(5, "Failed to load image
'/usr/local/share/marionnet/images/launcher-icons/marionnet-launcher.png': Fatal error in
PNG image file: IDAT: invalid distance too far back")
```

To correct, first install optipng, fix the error and restart Marionnet.

```
$ sudo apt-get install opting
$ optipng -quiet -fix /usr/local/share/marionnet/images/launcher-icons/marionnet-
launcher.png
$ sudo marionnet-daemon.byte
```

At this moment you may still get some warnings, like the ones shown in the picture below.



Make the following corrections (verify that the names of the files are without typos) by typing into your terminal window:

```
$ optipng -quiet -fix /usr/local/share/marionnet/images/ico.cable-serial.small.png
$ optipng -quiet -fix /usr/local/share/marionnet/images/ico.script-analyse-3.med.png
$ optipng -quiet -fix /usr/local/share/marionnet/images/marionnet-launcher.png
$ optipng -quiet -fix /usr/local/share/marionnet/images/ico.cable-serial-left-
right.small.png
$ optipng -quiet -fix /usr/local/share/marionnet/images/ico.uml-gray.small.png
$ optipng -quiet -fix /usr/local/share/marionnet/images/ico.dado.24.png
$ optipng -quiet -fix /usr/local/share/marionnet/images/ico.resource-2.med.png
$ optipng -quiet -fix /usr/local/share/marionnet/images/ico.dado-no.24.png
$ optipng -quiet -fix /usr/local/share/marionnet/images/ico.xml-16.inv.png
$ optipng -quiet -fix /usr/local/share/marionnet/images/ico.script-finish.med.png
$ optipng -quiet -fix /usr/local/share/marionnet/images/ico.software-gray.small.png
$ optipng -quiet -fix /usr/local/share/marionnet/images/ico.script-start-2.med.png
$ optipng -quiet -fix /usr/local/share/marionnet/images/ico.cable-serial.xxl.png
$ optipng -quiet -fix /usr/local/share/marionnet/images/ico.diffuser.orig.png
$ optipng -quiet -fix /usr/local/share/marionnet/images/ico.hardware-gray.small.png
$ optipng -quiet -fix /usr/local/share/marionnet/images/ico.cable-serial-left-
right.large.png
```

This solution is from https://bugs.launchpad.net/marionnet/+bug/1580349 and was proposed as a temporarily fix by JulioJu.

Restart Marionnet to verify that all the warnings disappeared.

```
$ sudo marionnet-daemon.byte
```

Your daemon should start now without any problems warning you that you are using it as root. This is not so important at this moment. We will handle it later in the section **Start Marionnet with a different user name and create a launcher**. If you reached this point you have some good progress so far. To check if a daemon is running we can look for it with:

```
$ ps –ef | grep <daemon process name>
```

and to start/stop/restart a daemon we do:

```
$ /etc/init.d/ <daemon process name> [start/stop/restart]
```

Use the commands above to check if your marionnet daemon is up and running.

Option #2: Configure system to run the *Marionnet daemon* on startup

The Marionnet wiki provides post-installation setup instructions that describe this procedure. Here are the steps I followed on this Kali Linux Light 2017 system:

To launch the *Marionnet daemon* from your startup scripts, add these lines to your */etc/rc.local* file (you need to do this as a super-user or as *root*):

```
$ sudo vi /etc/rc.local
```

Unfortunately, *rc.local* is considered deprecated and it is no longer activated on some new Linux distributions. I found a workaround solution. To fix this, first we have to create a ***rc-local.service*** file in ***/etc/systemd/system/***

```
$ sudo vi /etc/system/system/rc-local.service
```

The content of the file is

```
[Unit]
Description=/etc/rc.local Compatibility
ConditionPathExists=/etc/rc.local

[Service]
Type=forking
ExecStart=/etc/rc.local start
TimeoutSec=0
StandardOutput=tty
RemainAfterExit=yes
SysVStartPriority=99

[Install]
WantedBy=multi-user.target
```

After creating the file save it and let’s create the *rc.local* file in */etc/*

```
$ sudo vi /etc/rc.local
```

Then, add these lines in the file:

```
chmod a+rw /dev/net/tun
/usr/local/sbin/marionnet-daemon.byte &> /dev/null &
```

Save the file and make it executable

```
$ sudo  chmod +x /etc/rc.local
```

and enable the service on system boot

```
$ sudo  systemctl enable rc-local
```

You will need to reboot the host system before the daemon will run (or start it manually if you want to run it now). But you can wait until you complete the configuration changes listed below before restarting.

Ensure *X* is configured correctly

We need to ensure that *X* is working correctly so we can use *Wireshark* in later experiments.

First, check that the *X* server is listening to TCP or not. Marionnet uses TCP to communicate between virtual machine X clients and the host *X server*. Listening to TCP is disabled by default for security reasons. In my experience, the Marionnet install script did not make all the necessary changes to the configuration files.

Search for the *X* process and see if it is running with the *no listen tcp* option. The easiest way to do this is to look for the string, *no listen*, in the output of the *ps* command:

```
$ ps -ef | grep nolisten
root  1287  1273  1 10:39 tty7  00:00:03 /usr/bin/Xorg :0 -audit 0 -novtswitch -auth
/var/run/gdm3/auth-for-Debian-gdm-k5vPvh/database -nolisten tcp vt7
blinklet  2157  2097  0 10:45 pts/0  00:00:00 grep nolisten
```

If you see the *X* process running with the option: *-nolisten tcp*, as in the example above, then you need to fix the *gdm3* configuration and/or the *X server* configuration.

/etc/gdm3/daemon.conf

The display manager configuation scripts may need to be changed to allow X to use TCP. Depending on which Linux distribution you use, you may have a different display manager and different configuration files. In this case, we are using Kali Linux Light 2017.1-vbox-i686, which uses the *gdm3* display manager.

Edit the file */etc/gdm3/daemon.conf* (as superuser)

```
$ sudo vi /etc/gdm3/daemon.conf
```

In the file, enter the following text below the [security] line:

```
[security]
DisallowTCP=false
```

Save the file. You will need to restart the display manager so the changes will take effect (see the restart procedure later in this post) but, first, also check the X Server configuration file, */etc/X11/xinit/xserverrc* (see below).

/etc/X11/xinit/xserverrc

The *marionnet_from_scratch* script should update the */etc/X11/xinit/xserverrc* during the installation process. However, you should check the file to ensure the change was made and, if not, change the file as described below.

List the file contents. If you see the text, *nolisten tcp* in the file, edit the file */etc/X11/xinit/xserverrc* and remove the ***nolisten tcp*** text from the line, ***exec /usr/bin/X -nolisten tcp "$@".***

```
$ sudo vi /etc/X11/xinit/xserverrc
```

The new file should look like:

```
#!/bin/sh
exec /usr/bin/X "$@"
```

Restart the display manager

Now, you need to restart *gdm3* so that it will run allow *X* to connect to TCP but a simple restart of your system will not work. You need to do the following.

First, log out of your account using the Gnome menu command:

```
Menu → Logout
```

Then, press `Ctrl—Alt—F1` to switch to the text console.

Press the `Return` key to get a login prompt.

Log back in using your userid and password.

Then, enter the command:

```
$ sudo /etc/init.d/gdm3 restart
```

The system will ask you to log in again and the desktop should start up. If not, press `Ctrl—Alt—F7` to connect to the desktop sesion.

Now, the *gdm3* configuration is changed and the change will persist after restarting your machine.

Verify change

If you run the *ps* command again, you should see no processes running with the *-nolisten tcp* option.

```
$ ps -ef | grep nolisten
```

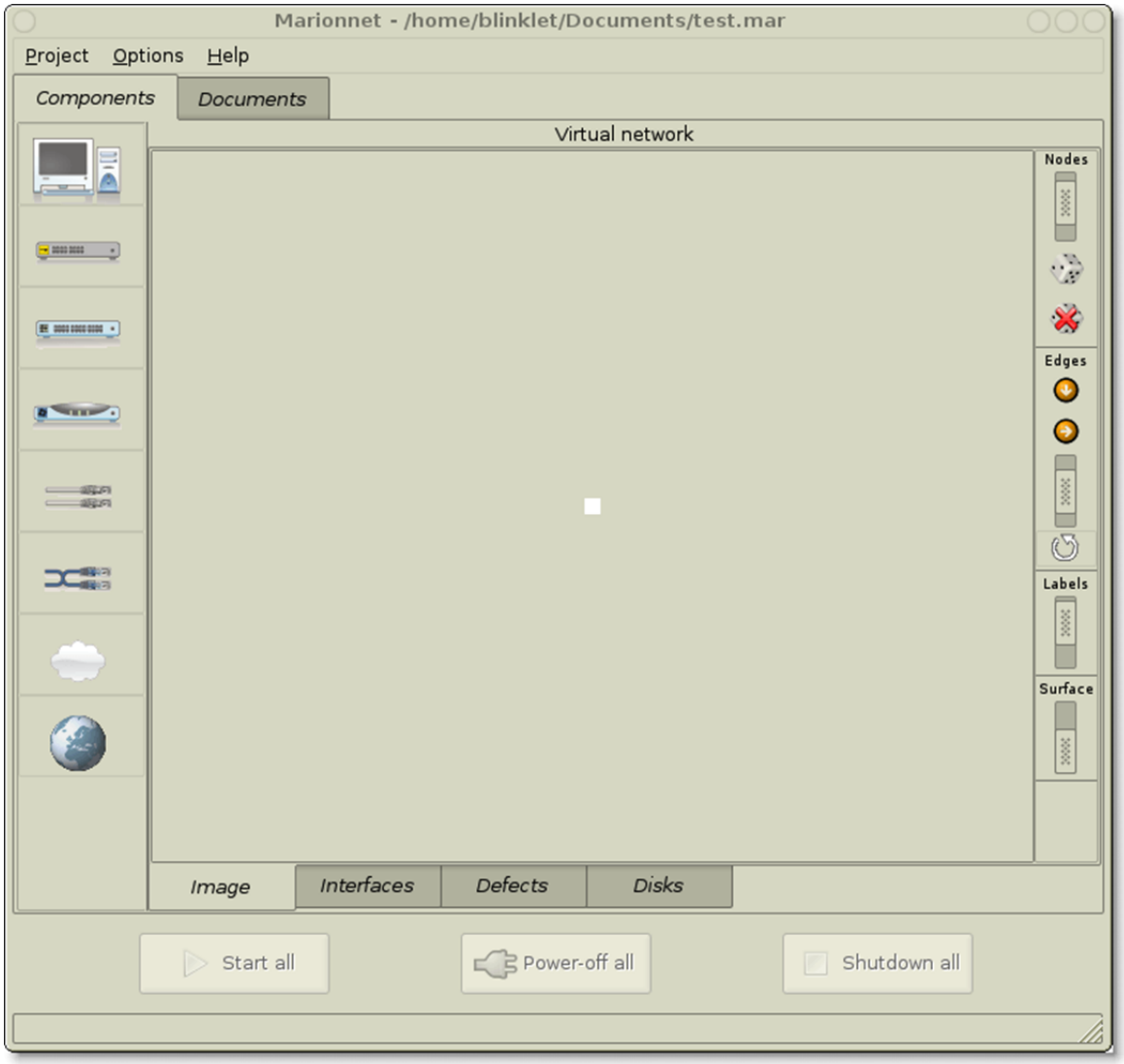
Start Marionnet

Start Marionnet by running the command:

```
$ marionnet.byte
```

A splash screen showing some information about the Marionnet software will appear. Click on the splash screen to make it go away and then you will see the Marionnet GUI.

Create a desktop shortcut(launcher) as shown in the section ***Start Marionnet with a different user name and create a launcher***.



To quit Marionnet, use the Marionnet menu command:

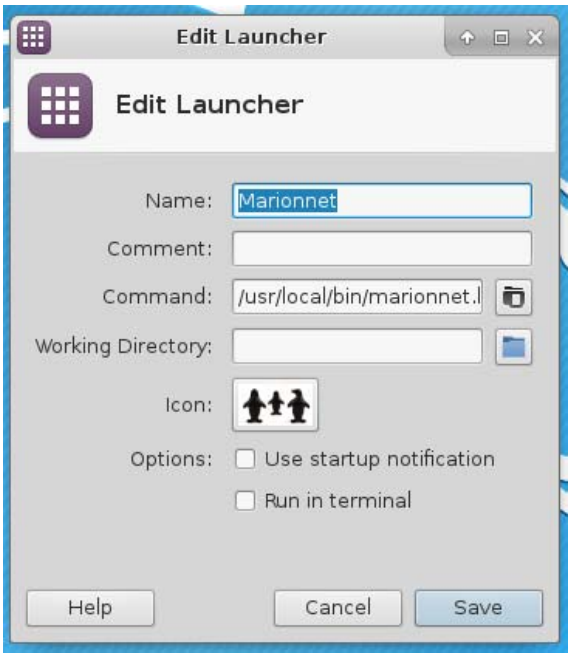
```
Project → Close
```

Start Marionnet with a different user name and create a launcher

Perform the following steps in order to create a new user and run Marionnet.

1. Open a terminal
2. Add a new user ***$ useradd –m username*** to create the home user directory
3. Create a password for the user ***$ passwd username***
4. Add user to the sudo group to allow the user to install soft, do upgrades…
   $ usermod –a –G sudo username
5. Change the default shell for the new user to bash ***$ chsh –s /bin/bash username***
6. ***$ logout*** in order to be able to login with the new use
7. After the GUI starts login with the new username and password
8. Right click on the screen to create a launcher
9. Fill in the launcher like in the picture below. (/usr/local/bin/marionnet.byte)
10. Download the icon from the internet and browse the icon object in the edit launcher until you get the downloaded icon.

11. The username=***userm*** and the password =***userm*** for the newly created account. You can use this account after you restart your Kali Linux. In the main screen double click on your Marionnet launcher icon to start marionnet.



Starting the VirtualBox machine

I have prepared a KaliLinux image that can be imported into VirtualBox. To do this, first you have to install VirtualBox from https://www.virtualbox.org/wiki/Downloads . Please download the most recent and stable version.

While installing the download you will be asked a series of questions. Answer with “Install” or “yes” to all this questions.

Also download the **VirtualBox 5.1.26 Oracle VM VirtualBox Extension Pack** as your already installed VirtualBox will give you an error first time you run KaliLinux virtual machine without the extension pack.

Now you can enjoy your KaliLinux with either the userm accout or the root account(username=***root***, password=***toor***).

Adjusting KaliLinux guest OS inside VirtualBox

Let’s assume that when you start KaliLinux inside VirtualBox the guest OS is not shown on the whole screen. It happened to me while I was trying to install Marionnet on Debian 9. To correct this issue follow the steps below:

1. Download VBoxGuestAdditions 5.1.26.iso from http://download.virtualbox.org/virtualbox/5.1.26/VBoxGuestAdditions_5.1.26.iso and mount the iso image
2. Create a copy of the image anywhere on your hard drive inside KaliLinux. I did this in my home folder where I created a folder named ***VBoxGuestAdd***.
3. In a terminal window:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install dkms
```

4. Go to your newly created folder VBoxGuestAdd and change the permissions for the corresponsing linux additions file. In my case:

```
$ sudo chmod 777 VBoxLinuxAdditions.run
```

5. Execute

```
 $ sudo ./VBoxLinuxAdditions.run
```

6. Restart your KaliLinux system and you should see your OS guest on the whole screen