How to securely isolate Damn Vulnerable Linux with VirtualBox

Motivation

I have this vision of going through and testing software inside of a closed lab that allows me to understand exploits and vulnerabilities at a deep level. Instead of starting from scratch and rolling my own distribution of Linux, I chose to use Damn Vulnerable Linux. This is to save time and get running as quickly as possible. The key here is that all of the virtual machines will run inside of the VirtualBox network of Host-only. This allows safe handling (isolation) of systems that could otherwise be exploited and leveraged as jump points into your corporate or home infrastructure.

Test environment layout

My workstation is running Ubuntu 16.10.
I am first installing VirtualBox 5.1.6 for Ubuntu, using method 2 below.
Then Damn Vulnerable Linux 1.5.

Install VirtualBox

Method 1:

Download software package from:

https://www.virtualbox.org/wiki/Linux_Downloads

```
$ cd ~/Downloads
$ wget http://download.virtualbox.org/virtualbox/5.1.10/virtualbox-
5.1_5.1.10-112026~Ubuntu~yakkety_amd64.deb
$ sudo dpkg -i virtualbox-5.1_5.1.10-112026-Ubuntu-yakkety_amd64.deb
```

Method 2:

Append the following line to your /etc/apt/sources.list (assuming it doesn't exist):

```
deb http://download.virtualbox.org/virtualbox/debian yakkety contrib
```

From Terminal (the $ means you are running this from your regular user account):

```
$ cd ~/Downloads
$ wget https://www.virtualbox.org/download/oracle_vbox_2016.asc
$ wget https://www.virtualbox.org/download/oracle_vbox.asc
$ sudo apt-key add oracle_vbox_2016.asc
$ sudo apt-key add oracle_vbox.asc
$ sudo apt-get update
$ sudo apt-get install virtualbox
$ sudo apt-get install dkms
$ sudo apt install virtualbox-ext-pack
$ sudo apt-get install virtualbox-ext-pack virtualbox-guest-additions-iso
```

Either method will work, I prefer method 2 because then you can get updates with the software.

Configure the multiple network domains:

This assumes you have not done this or need to modify your network.

Create NAT'ed Network:

When you see the "# at the beginning of a command line, that means you are running as root.

```
# VBoxManage natnetwork add \
  --netname 192.168.139-NAT \
  --network "192.168.139.0/24" \
  --enable --dhcp on
```

Create the DHCP server:

```
# VBoxManage dhcpserver add \
  --netname 192.168.139-NAT \
  --ip 192.168.139.3 \
  --lowerip 192.168.139.101 \
  --upperip 192.168.139.254 \
  --netmask 255.255.255.0 \
  --enable
```

Create hostonly interface:

```
# VBoxManage hostonlyif create
# VBoxManage hostonlyif ipconfig vboxnet0 \
  --ip 172.20.0.1 \
  --netmask 255.255.255.0
# VBoxManage dhcpserver add \
  --ifname vboxnet0 \
  --ip 172.20.0.3 \
  --lowerip 172.20.0.101 \
  --upperip 172.20.0.254 \
  --netmask 255.255.255.0
# VBoxManage dhcpserver modify \
  --ifname vboxnet0 \
  --enable
```

To list the NAT'ed networks:

```
# VBoxManage list natnetworks
```

Output:

```
NetworkName:    192.168.139-NAT
IP:             192.168.139.1
Network:        192.168.139.0/24
IPv6 Enabled:   No
IPv6 Prefix:    fd17:625c:f037:a88b::/64
DHCP Enabled:   Yes
Enabled:        Yes
loopback mappings (ipv4)
        127.0.0.1=2
```

To List the DHCP server(s):

```
# VBoxManage list dhcpservers
```

Output:

```
NetworkName:    192.168.139-NAT
IP:             192.168.139.3
NetworkMask:    255.255.255.0
lowerIPAddress: 192.168.139.101
upperIPAddress: 192.168.139.254
Enabled:        Yes

NetworkName:    HostInterfaceNetworking-vboxnet0
IP:             172.20.0.3
NetworkMask:    255.255.255.0
lowerIPAddress: 172.20.0.101
upperIPAddress: 172.20.0.254
Enabled:        Yes

NetworkName:    HostInterfaceNetworking-vboxnet1
IP:             0.0.0.0
NetworkMask:    0.0.0.0
lowerIPAddress: 0.0.0.0
upperIPAddress: 0.0.0.0
Enabled:        No
```

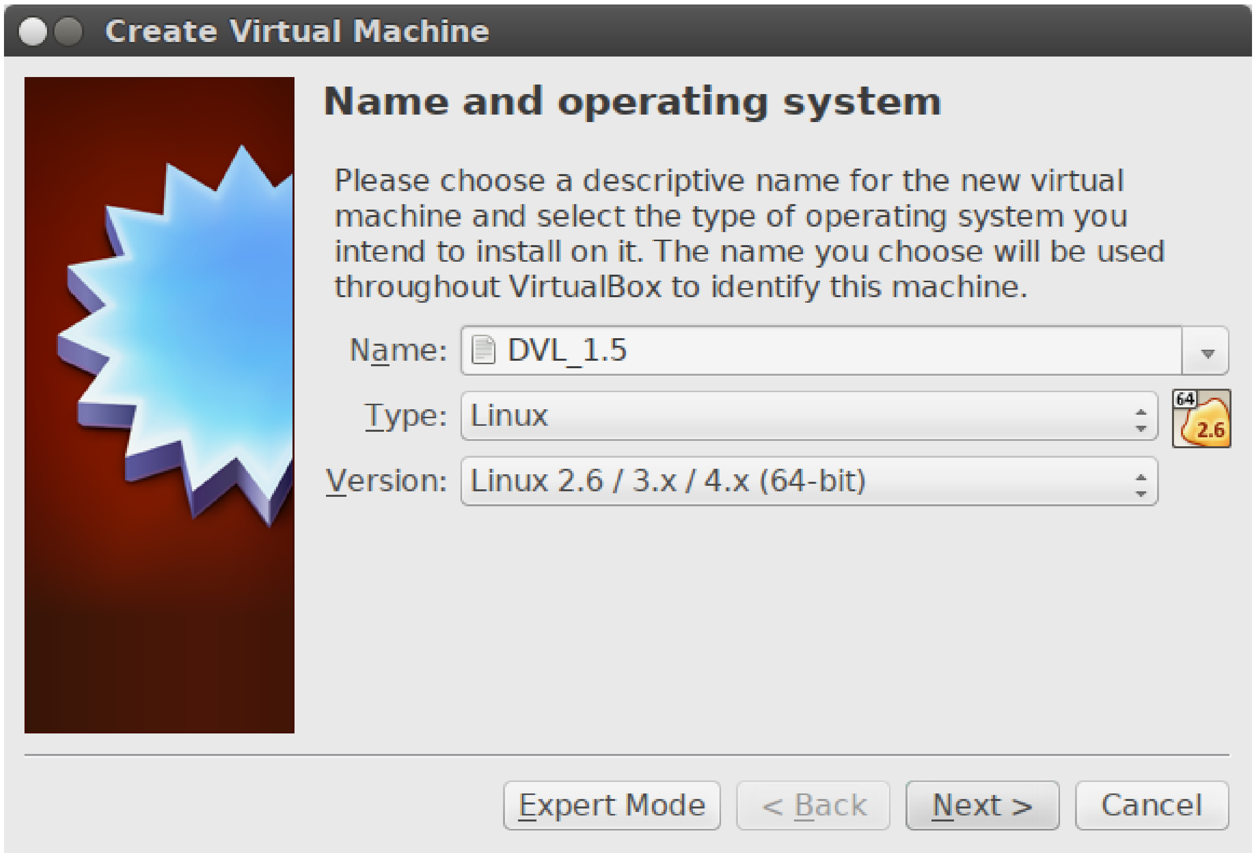
Get the distribution

Get the distrobution:

```
$ cd ~/Downloads/
$ wget
http://osdn.net/projects/sfnet_virtualhacking/downloads/os/dvl/DVL_1.5_Infect
ious_Disease.iso/
```

Configure a new Virtual Machine for DVL

Open virtualbox and create a new virtual machine.



Name and operating system

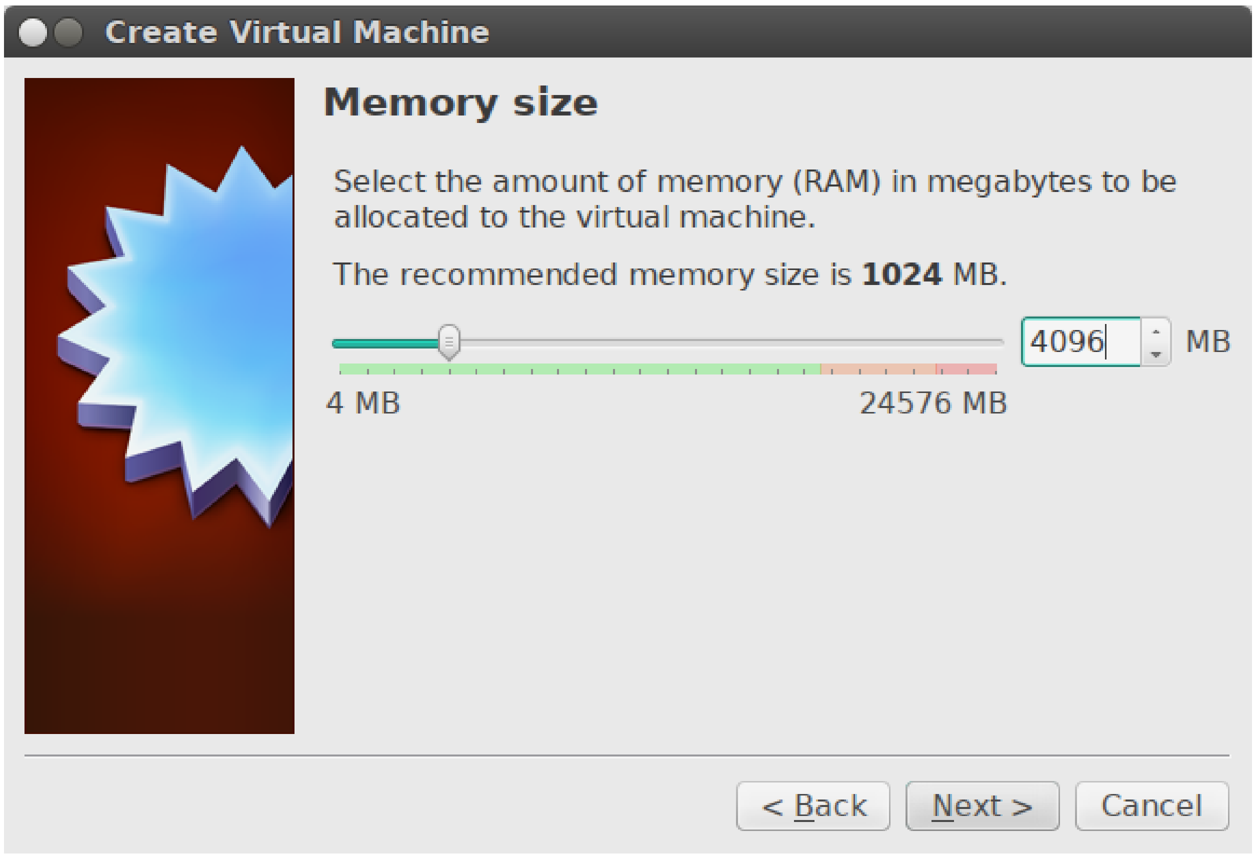
Please choose a descriptive name for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name: DVL_1.5

Type: Linux

Version: Linux 2.6 / 3.x / 4.x (64-bit)

I named my virtual machine “DVL_1.5”, choose what you wish here.


Create Virtual Machine

Memory size

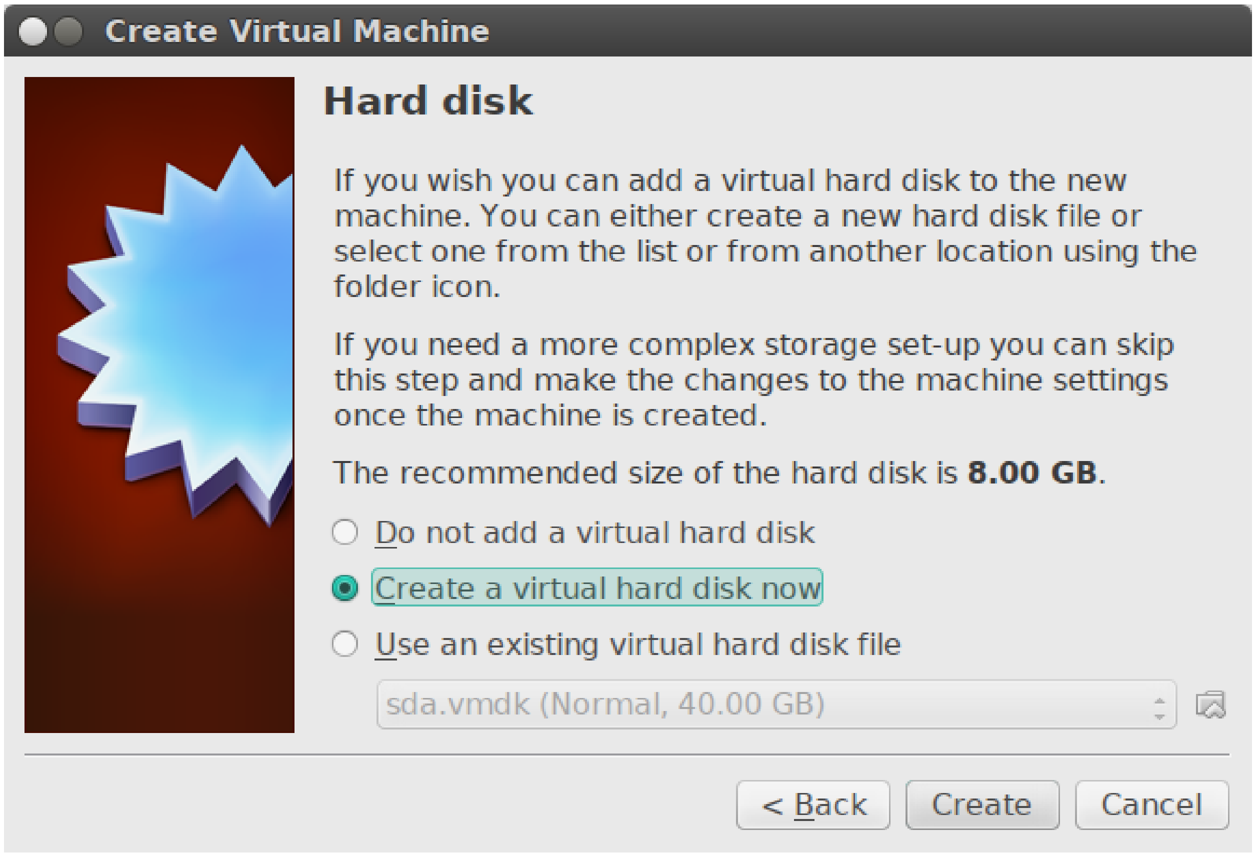
Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **1024** MB.

4096 MB

4 MB 24576 MB

I would recommend 4GB of memory. You can use less, but if you have the extra memory, I highly recommend giving the system more than a fair amount of memory.



Create Virtual Machine

Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

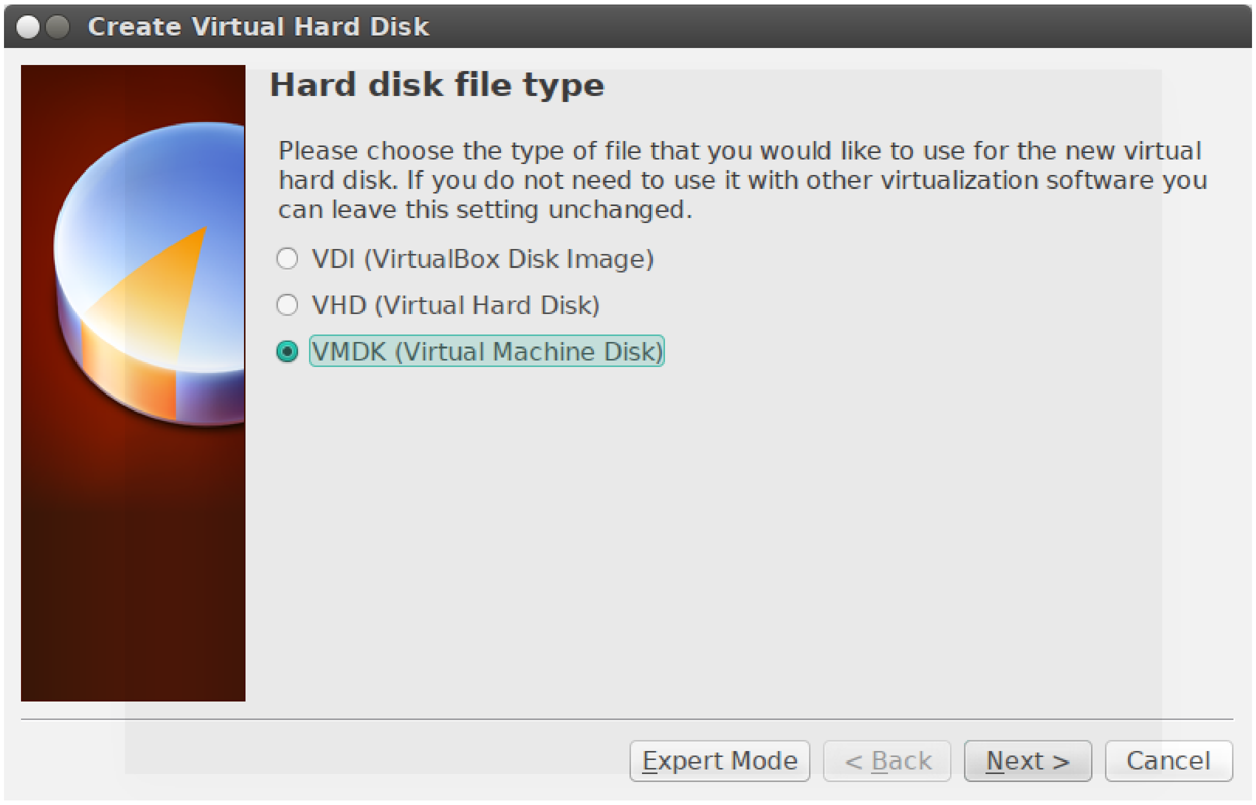
If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the hard disk is **8.00 GB**.

- Do not add a virtual hard disk
- [✓ Create a virtual hard disk now]
- Use an existing virtual hard disk file

sda.vmdk (Normal, 40.00 GB)

Check the create hard drive radio button.

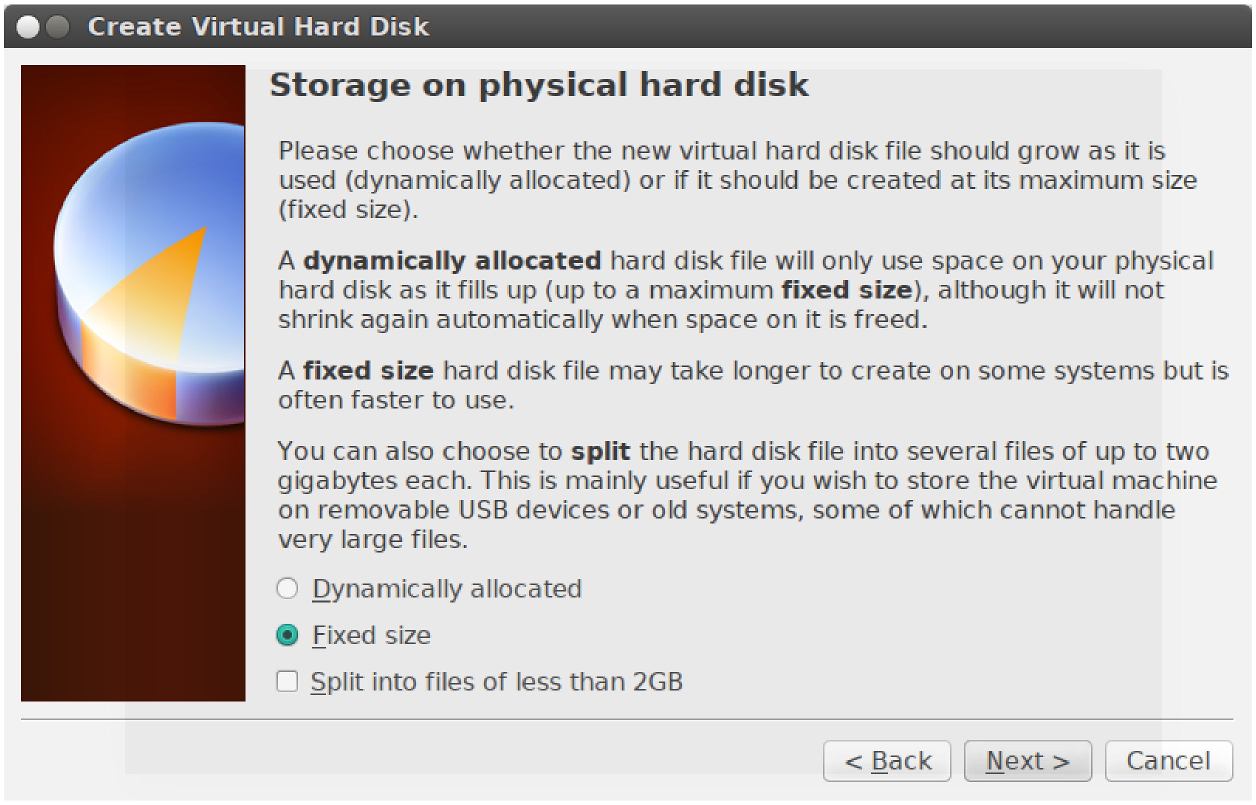

Create Virtual Hard Disk

Hard disk file type

Please choose the type of file that you would like to use for the new virtual hard disk. If you do not need to use it with other virtualization software you can leave this setting unchanged.

- VDI (VirtualBox Disk Image)
- VHD (Virtual Hard Disk)
- [✓ VMDK (Virtual Machine Disk)]

Choose the VMDK radio button.



Storage on physical hard disk

Please choose whether the new virtual hard disk file should grow as it is used (dynamically allocated) or if it should be created at its maximum size (fixed size).

A **dynamically allocated** hard disk file will only use space on your physical hard disk as it fills up (up to a maximum **fixed size**), although it will not shrink again automatically when space on it is freed.

A **fixed size** hard disk file may take longer to create on some systems but is often faster to use.

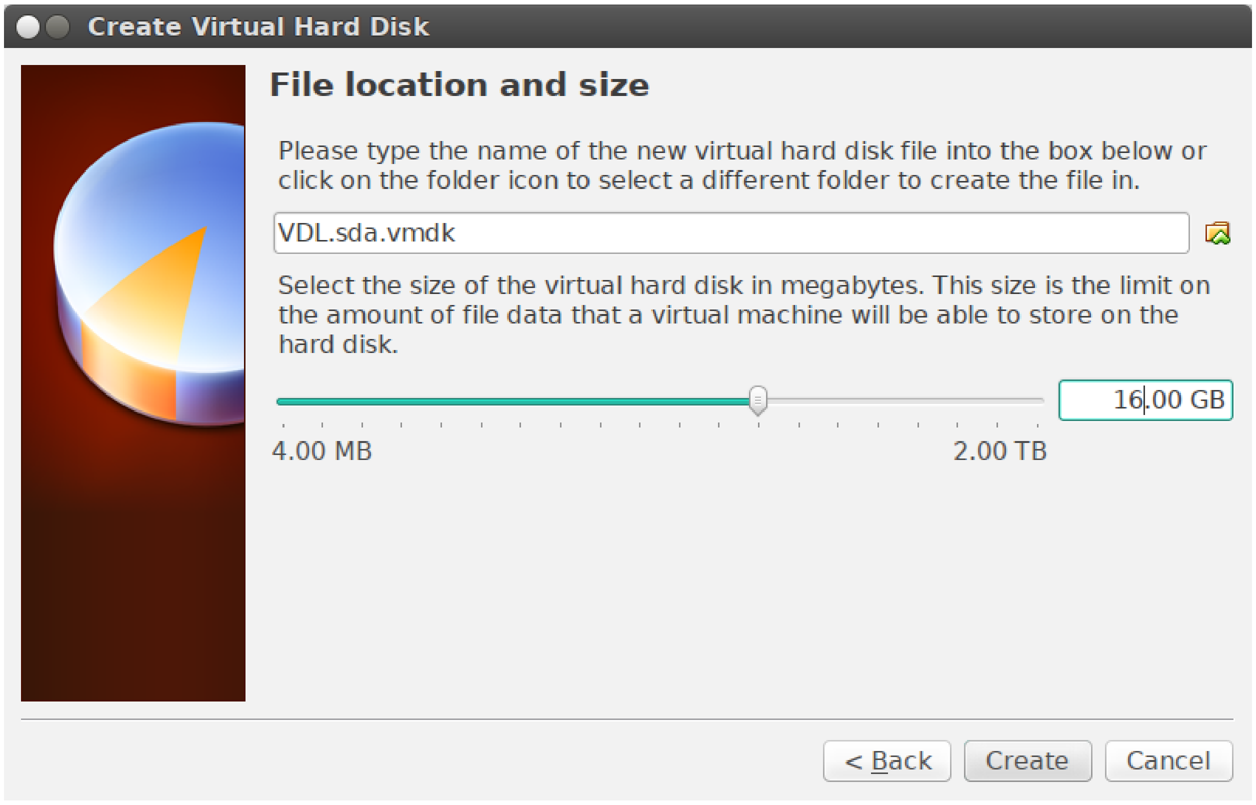
You can also choose to **split** the hard disk file into several files of up to two gigabytes each. This is mainly useful if you wish to store the virtual machine on removable USB devices or old systems, some of which cannot handle very large files.

○ Dynamically allocated

● Fixed size

☐ Split into files of less than 2GB

Choose the Fixed size radio button.



File location and size

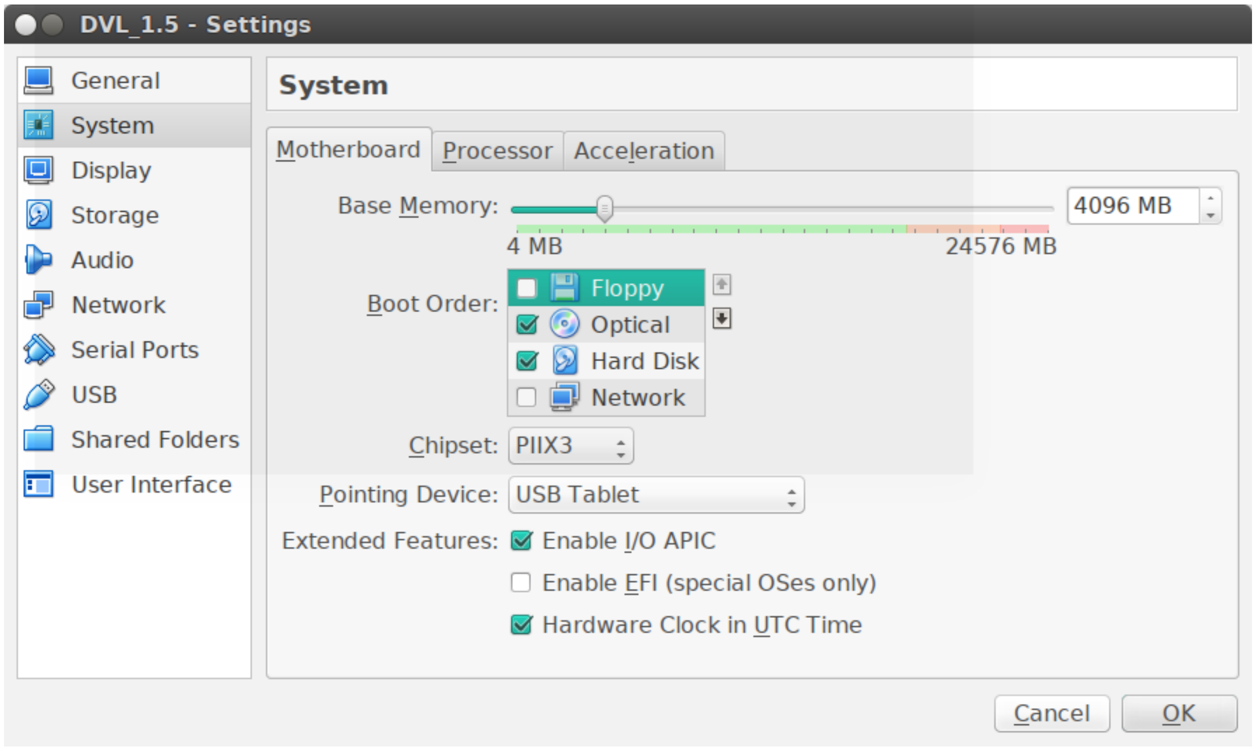
Please type the name of the new virtual hard disk file into the box below or click on the folder icon to select a different folder to create the file in.

VDL.sda.vmdk

Select the size of the virtual hard disk in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard disk.

16.00 GB

I named my disk, "VDL.sda.vmdk" and changed the size from 8GB to 16GB. The sda is so I know which disk this is, in case I want to add more later. This ties to the internal mapping of drives so I know which is which.


DVL_1.5 - Settings

General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface

System

Motherboard | Processor | Acceleration

Base Memory: 4096 MB (4 MB – 24576 MB)

Boot Order:
- [ ] Floppy
- [✓] Optical
- [✓] Hard Disk
- [ ] Network

Chipset: PIIX3

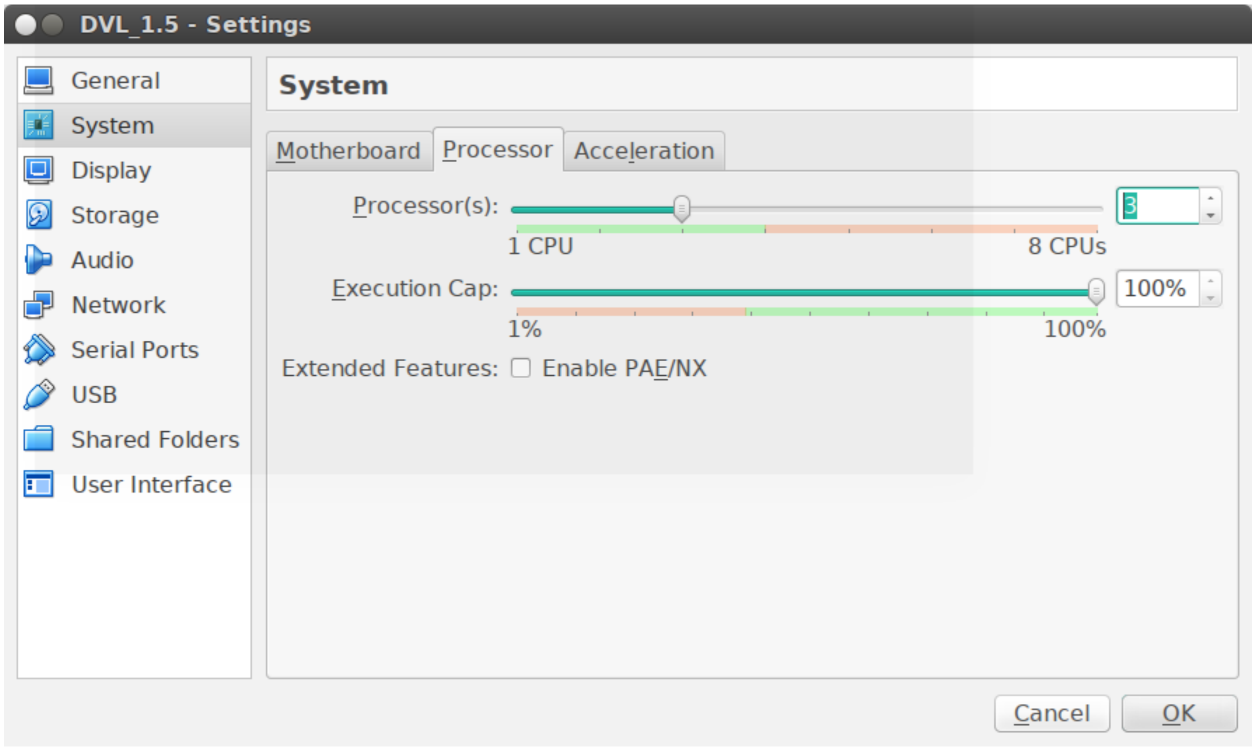
Pointing Device: USB Tablet

Extended Features:
- [✓] Enable I/O APIC
- [ ] Enable EFI (special OSes only)
- [✓] Hardware Clock in UTC Time

Cancel | OK

Change the memory to 4GB, if needed.

Uncheck the Floppy drive, don’t need it.



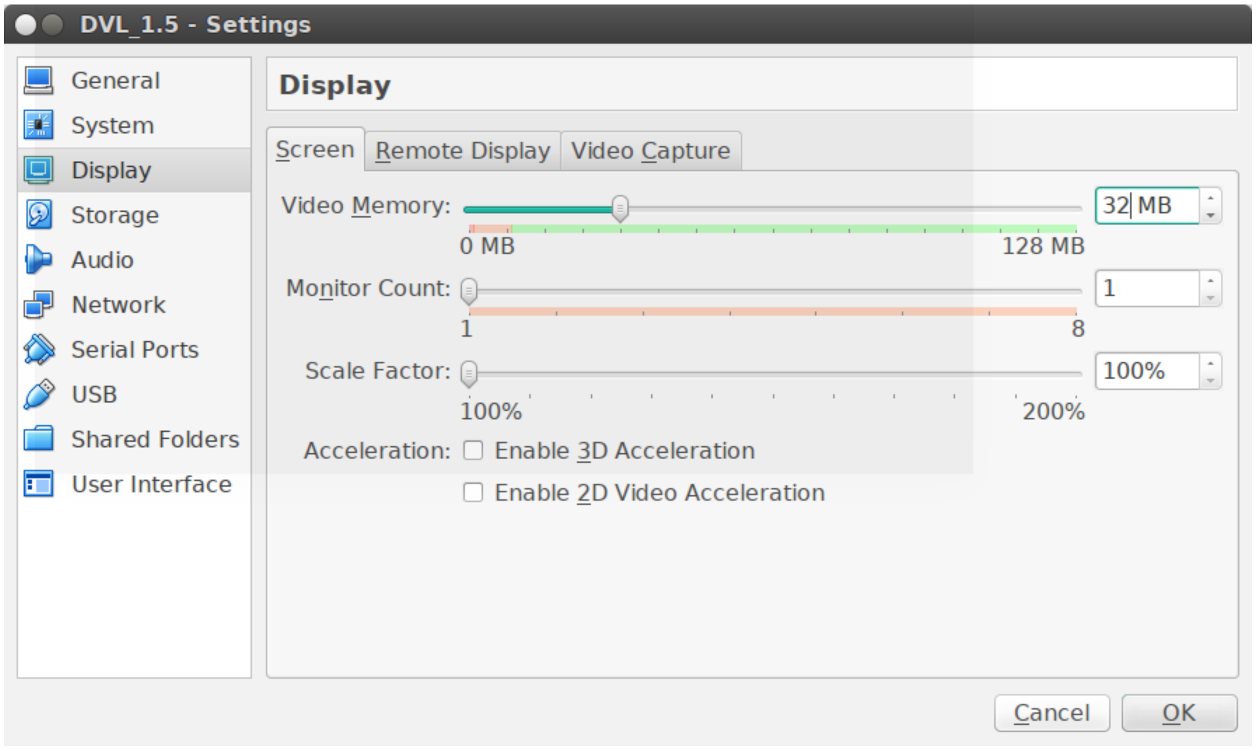
System

Processor(s): 3 (1 CPU – 8 CPUs)

Execution Cap: 100% (1% – 100%)

Extended Features: ☐ Enable PAE/NX

Change the processor count to “3”.



Display

Video Memory: 32 MB (0 MB – 128 MB)

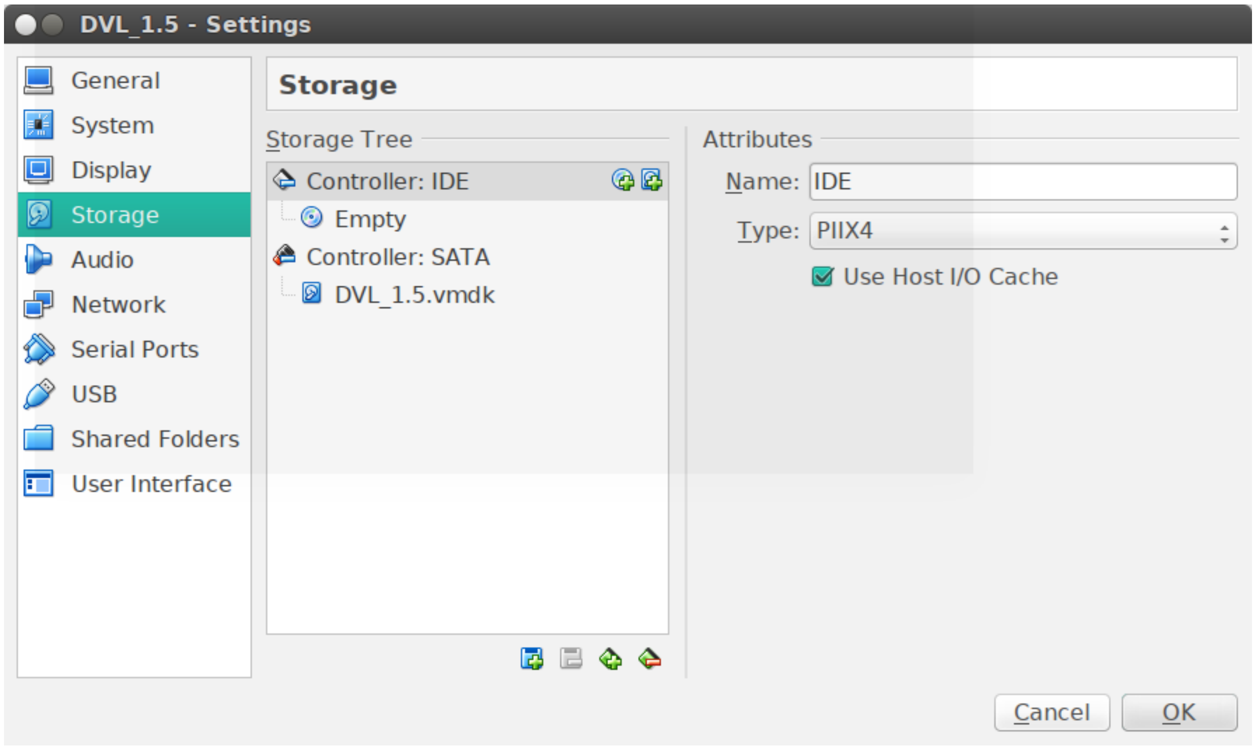
Monitor Count: 1 (1 – 8)

Scale Factor: 100% (100% – 200%)

Acceleration: ☐ Enable 3D Acceleration

☐ Enable 2D Video Acceleration

Change the Video Memory to 32 MB.



Storage

Storage Tree

Controller: IDE
Empty
Controller: SATA
DVL_1.5.vmdk

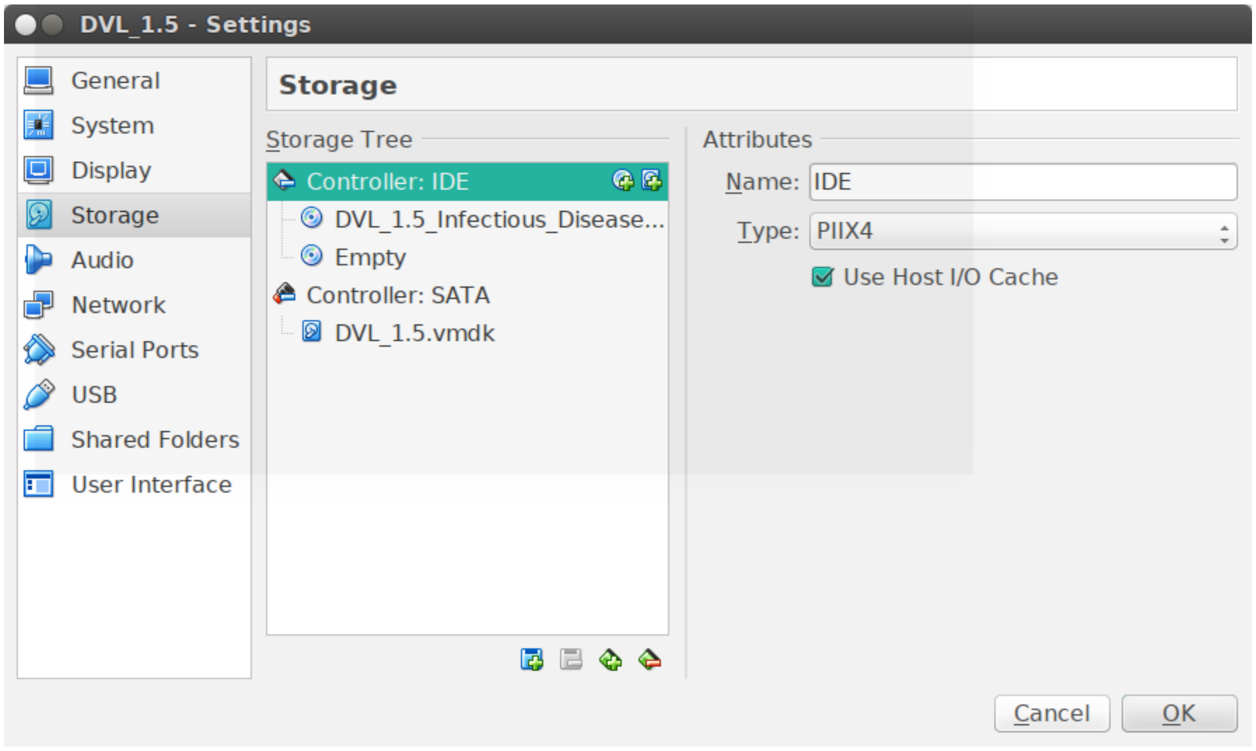
Attributes

Name: IDE

Type: PIIX4

✓ Use Host I/O Cache

On the Controller: IDE, click on the little plus button over the CDROM icon. Point back to the ISO image for DVL 1.5.


DVL_1.5 - Settings

General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface

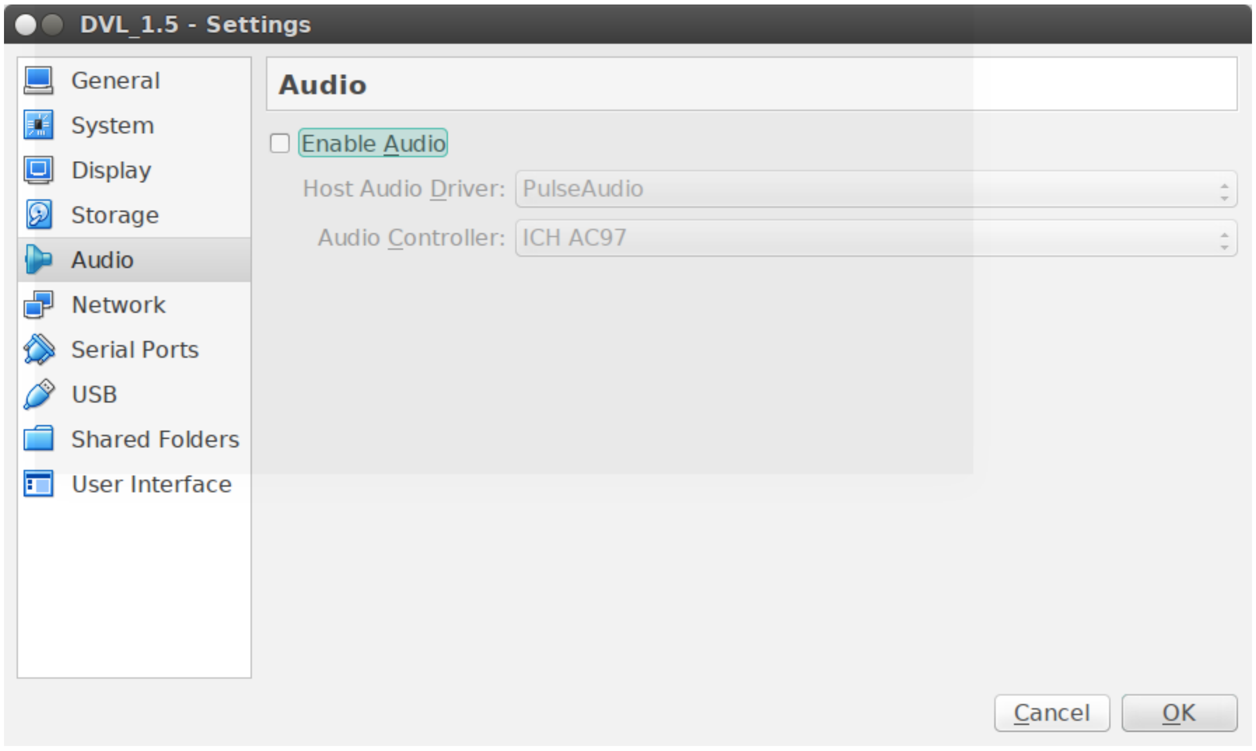
Storage

Storage Tree
Controller: IDE
DVL_1.5_Infectious_Disease...
Empty
Controller: SATA
DVL_1.5.vmdk

Attributes
Name: IDE
Type: PIIX4
[✓] Use Host I/O Cache

Cancel OK

With the DVL image mounted.


DVL_1.5 - Settings

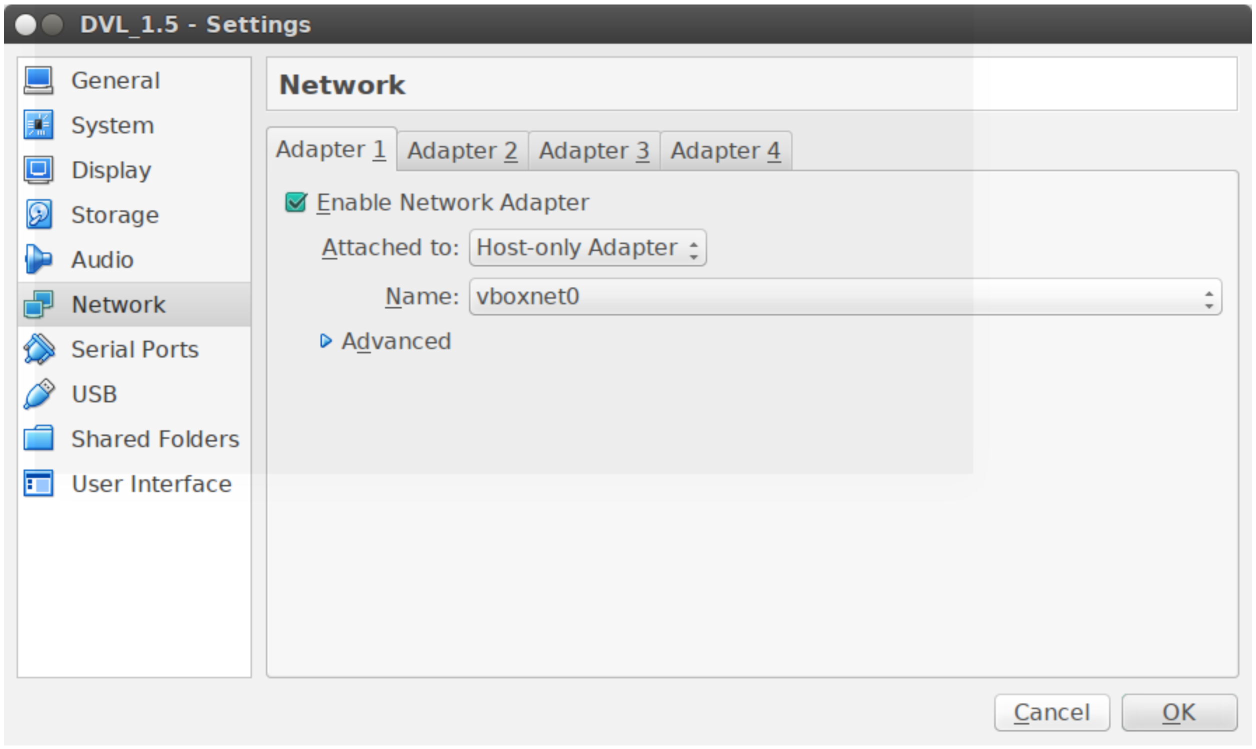
General
System
Display
Storage
Audio
Network
Serial Ports
USB
Shared Folders
User Interface

Audio

[ ] Enable Audio
Host Audio Driver: PulseAudio
Audio Controller: ICH AC97

Cancel OK

I like to disable the Audio, it's really not needed.



Enable Network Adapter ✓

Attached to: Host-only Adapter

Name: vboxnet0

Finally, on the network tab, point the network interface to:
Attached to: Host-only Adapter
Name: vboxnet0

Setup and deployment of Vulnerable Damn Linux

Start the virtual machine instance of DVL 1.5, at the boot: prompt, hit Enter.

Login with:

Username:	root
Password:	toor

Check disk format, should be /dev/sda:

```
# fdisk -l
```

Output:

```
Disk /dev/sda: 17.1 GB, 17179869184 bytes
255 heads, 63 sectors/track, 2088 cylinders
```

```
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start          End      Blocks   Id  System
```

Begin the process of formatting the disk:

```
# fdisk -l
```

Input (in this sequence):

```
: m
: n
: p
: 1
: Enter-Key for default
: Enter-Key again for default
```

With the 16GB disk, I had 2088 sectors.

View the new partion (inside of fdisk):

```
Disk /dev/sda: 17.1 GB, 17179869184 bytes
255 heads, 63 sectors/track, 2088 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start          End      Blocks   Id  System
/dev/sda1           2088           2088     8032+    83  Linux
```

Select “q” for quit:

```
# q
```

Format the partition on /dev/sda:

```
# mkfs.ext3 /dev/sda
# y
```

Create a new folder to mount the new partition:

```
# mkdir /mnt/dvl
```

Mount the partition:

```
# mount /dev/sda /mnt/dvl
```

Start the window system:

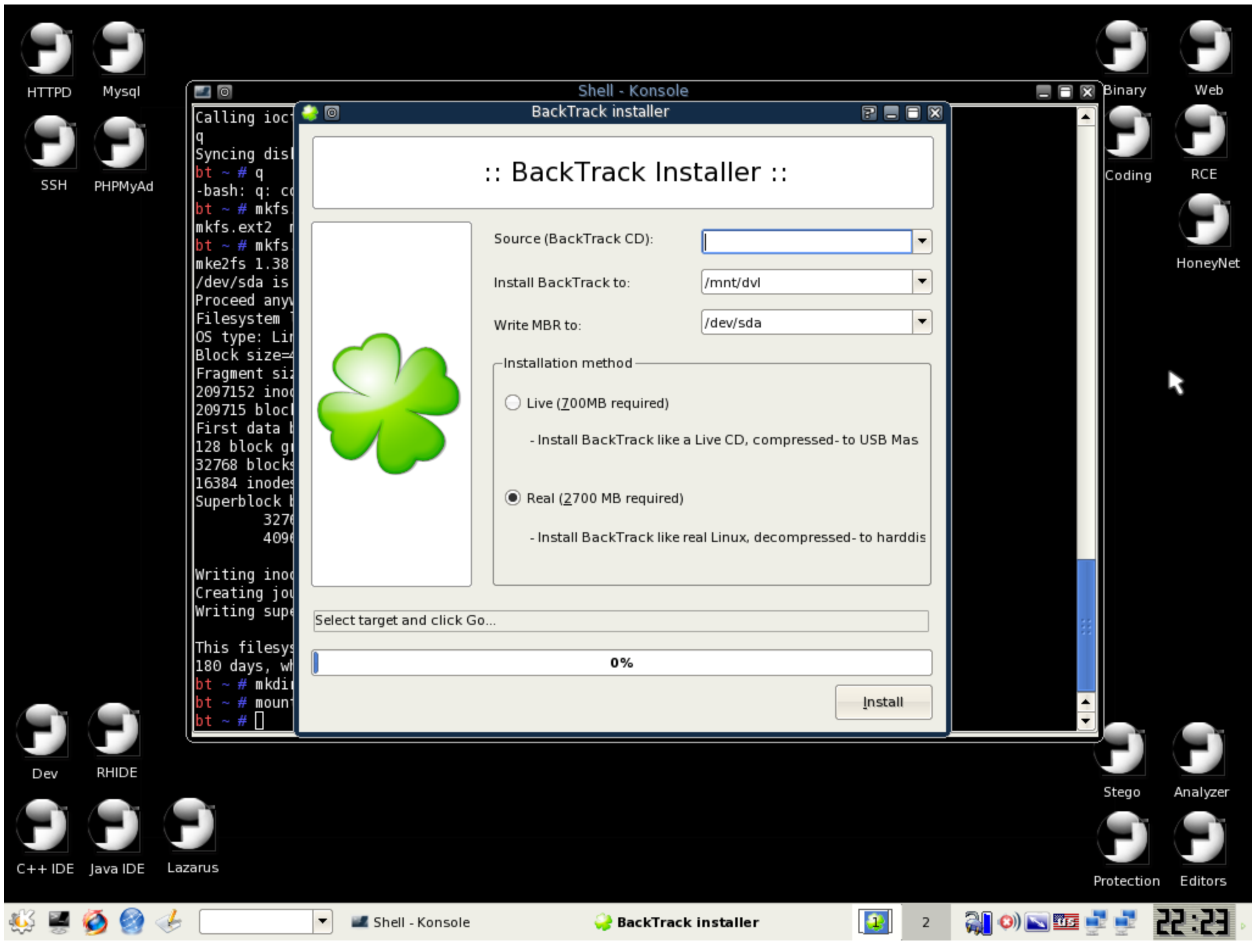
```
# startx &
```

In the bottom left of the screen, click on the KDE gear, an menu will pop up.

Go up on the menu and hover over "Be ReSlaxed", a new menu will appear.

Go right and hover over System, a new menu will appear.

Click on "BackTrack Installer".



```
Calling ioc
q
Syncing disl
bt ~ # q
-bash: q: co
bt ~ # mkfs
mkfs.ext2  r
bt ~ # mkfs
mke2fs 1.38
/dev/sda is
Proceed any
Filesystem
OS type: Lin
Block size=
Fragment siz
2097152 inod
209715 block
First data b
128 block gr
32768 blocks
16384 inodes
Superblock b
        3276
        4096

Writing inod
Creating jou
Writing supe

This filesys
180 days, wh
bt ~ # mkdi
bt ~ # moun
bt ~ #
```

:: BackTrack Installer ::

Source (BackTrack CD):
Install BackTrack to: /mnt/dvl
Write MBR to: /dev/sda

Installation method
Live (700MB required)
- Install BackTrack like a Live CD, compressed- to USB Mas
[✓ Real (2700 MB required)]
- Install BackTrack like real Linux, decompressed- to harddis

Select target and click Go...

0%

A new window will appear, labeled BackTrack Installer.

Leave the field for "Source (BackTrack CD)" Blank, that means empty.

The value for "Install BackTrack to" should auto populate with: "/mnt/dvl"

The value for "Write MBR to" should auto populate with: "/dev/sda"

Check the radio box for "Real (2700 MB required)".

Finally click on Install.

Once the installer hits 100%, click on the Close button.

Open a new terminal.

Run a chroot environment for the new install and install lilo.

NOTE: You might see an error with chroot, this is ok.

```
# chroot /mnt/dvl /bin/bash
# lilo -v
```

Stop the system, inside the terminal:

```
# exit
# shutdown -h now
```

In VirtualBox, open up Settings for the DVL_1.5 virtual machine and dis-connect the ISO file from the cdrom.

Start the virtual machine with VirtualBox.

Log into the virtual machine:

Login with:

Username:	root
Password:	toor

Backup image once built.

I used (inside my directory for virtual machines):

```
$ GZIP="-9"; export GZIP; time tar czf DVL_1.5.fresh.tgz DVL_1.5
```

It took over 11 minutes to compress the image with the highest compression, but the end result is a 2.1 GB gzip'ed tarbal that I can quickly recover from (I plan on dorking up the instance, so I need a recovery point – and yes, I know about Snapshot technology – sometimes a simple baseline is the best thing to recover from because the slate is pure; besides Snapshots are wonderful for quick change sets, but having a pure baseline to go back to helps when you have so many compounded changes that you don't remember at what point you want to revert to → which also speaks to having good labels on your snapshots). This is one of my quirks, I like using both methods (the tarball and Snapshots).

Conclusion

By following these steps, you will have a running instance of Vulnerable Damn Linux. From here, start playing with turning on SSHD so that you can remotely connect (from your workstation into said instance) and start configuration, or mis-configuration for other tools to exploit.