GRUB

Notes on grub, because it's grand :)

To apply *any* of these changes, make sure to run the following command after saving the configurations.

```
sudo update-grub
```

When you open the `/etc/defaults/grub` file for editing in the next section, the header comment will remind you of this. Don't overlook it!

```
sudo head /etc/default/grub


# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'
```

You can run the `info -f grub -n 'Simple configuration'` command to view the GRUB manual, starting at the `Simple configuration` section, or you can view the same manual from your web browser - it's up to you.

/etc/defaults/grub

Create a backup of your configuration before making any edits, so you can restore the original configuration if things get out of hand.

```
sudo cp /etc/defaults/grub /etc/defaults/grub.bak
```

The settings below will save last selected boot option as the default for next boot. To override it, simply reboot and select a different option.

```
GRUB_DEFAULT=saved
GRUB_SAVEDEFAULT=true
GRUB_TIMEOUT_STYLE=menu
GRUB_TIMEOUT=5
```

GRUB Console

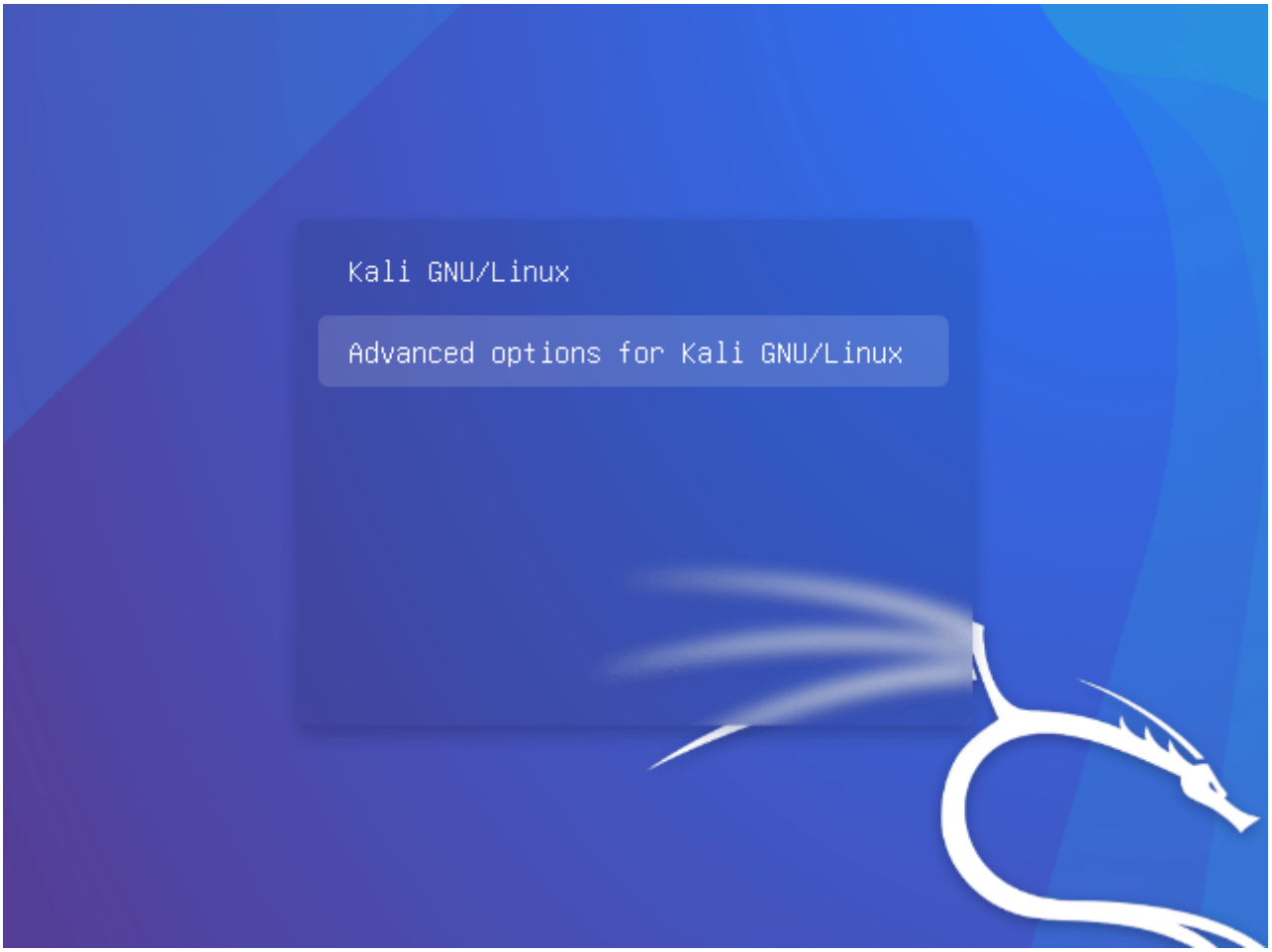
You may one day need to boot from a GRUB console, and to do this you'll need to select your kernel and initrd (Initialize Ram Disk) image manually. This might sound intimidating, but it really is not that bad and if you know where your MBR is located you should be able to accomplish this in just a few commands.

The file we want to use for our linux kernel on this machine is `vmlinuz-5.15.0-kali3-amd64` - yours might have a slightly different name, but it should begin with `vmlinuz`

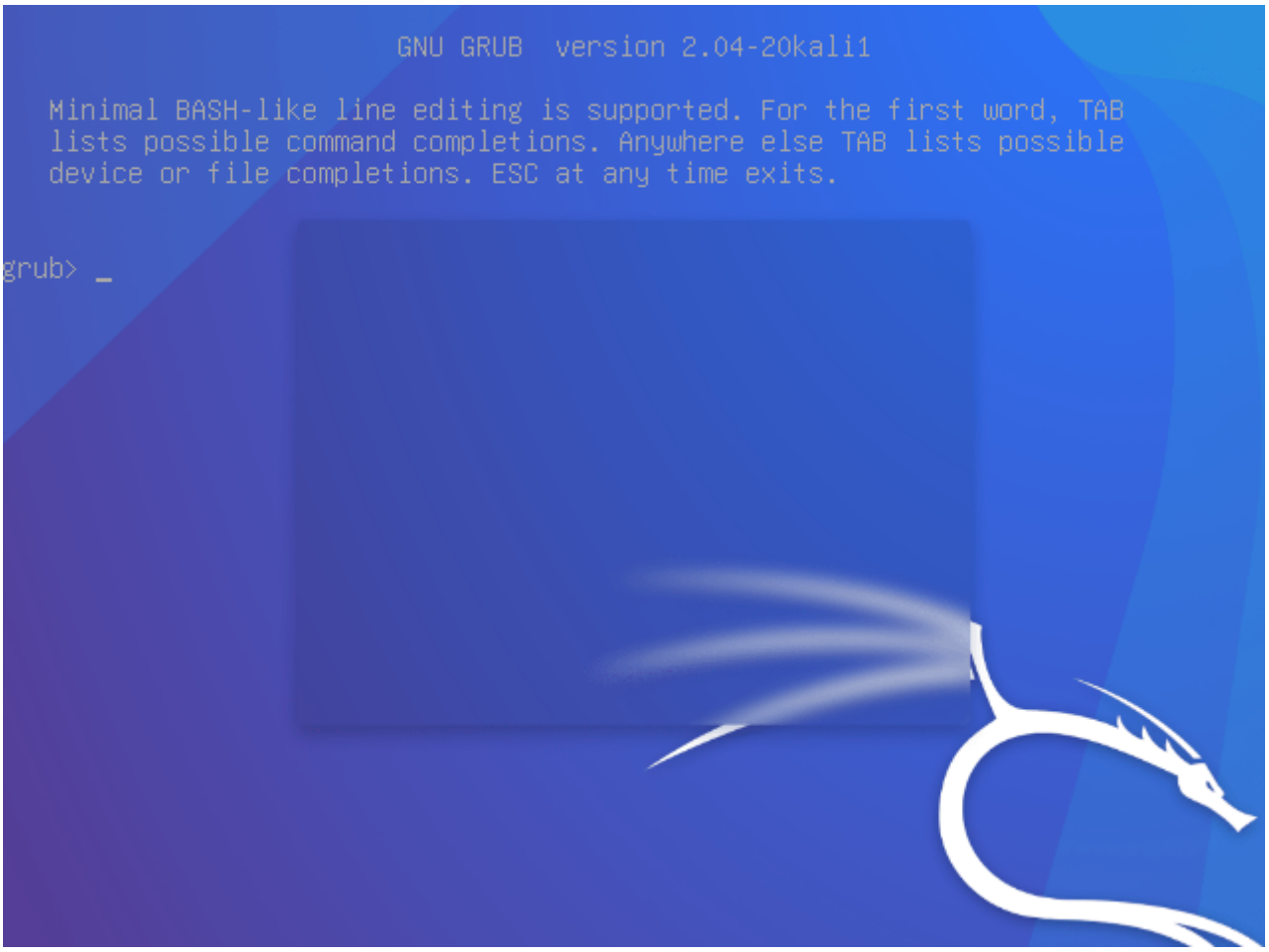
The file we want to use for our `initrd` image is named `initrd.img-5.15.0-kali3-amd64`, and again yours should be similar and begin with `initrd`.

Both of these files should exist in the `/boot/` directory of the MBR

On a Kali VM GRUB looks like this during boot. Simply interrupt the boot by pressing the up or down arrow on your keyboard to prevent the timeout from triggering and the system will not automatically proceed, it will wait for your input.

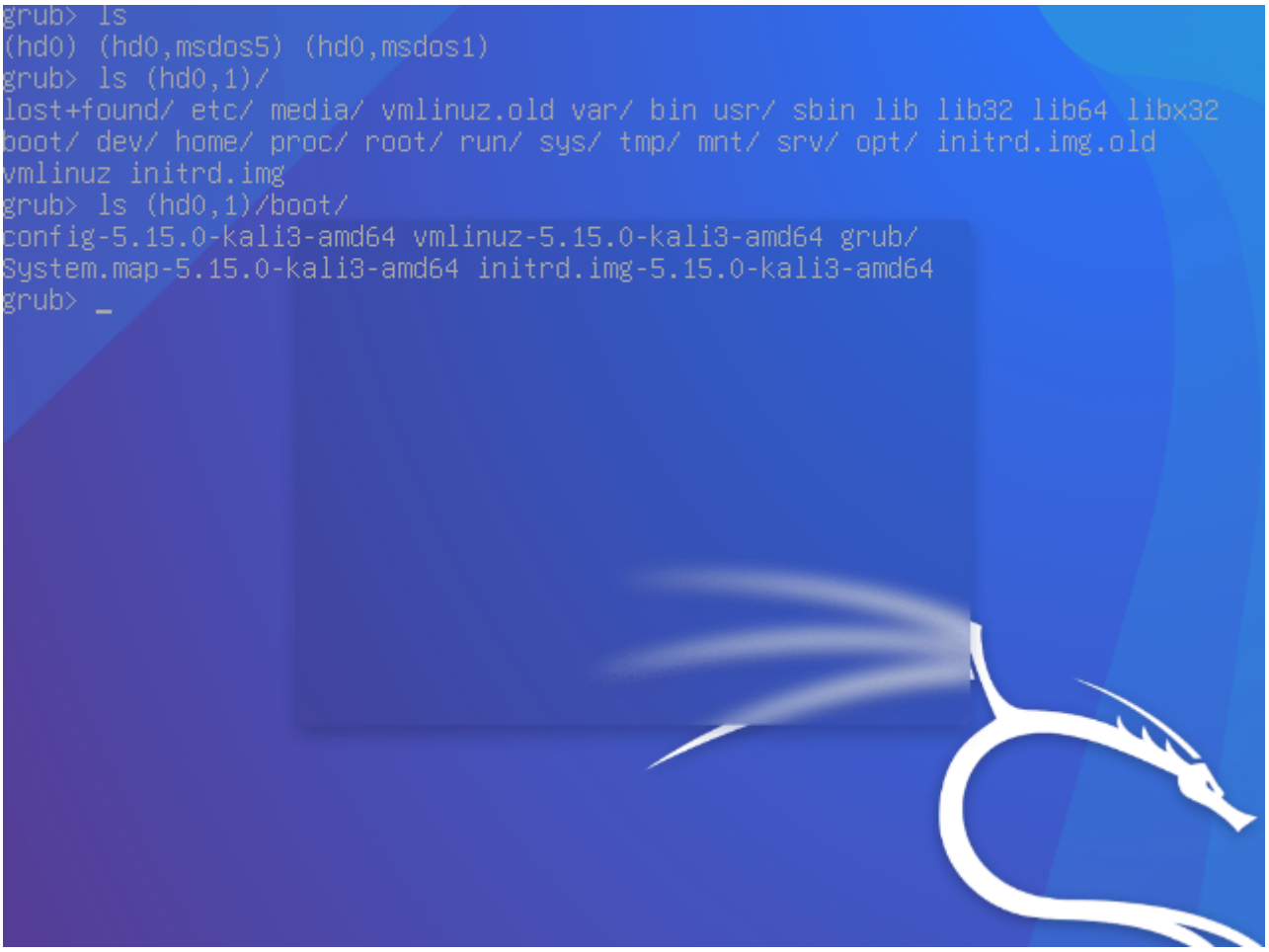


Now we can press `c` to enter the GRUB console, and you can practice booting from GRUB manually. You will see the console below.



In the next image, we ran the `ls` command, and seen the available filesystems output to the console. Since we are in a GRUB console, we have not yet set our root filesystem. If you have a lot of storage devices attached to your machine, this list might be much longer. In any case, the MBR should exist on the first sector of the drive that has your OS installed on it. First find the drive you want to boot from, then we can check the `/boot/` directory within the first partiton of this drive for the kernel and image file we need to boot the system.

For this output, it's clearly the `(hd0)` device since it is the only device available. Note that `(hd0)` is the device itself, and not the first sector. To view the contents in the first sector of the `(hd0)` device before we set it to be our root filesystem, we can run `ls (hd0,1)/`. You can leave out the `mdos` section, or you can type it out - both will have the same result. On some systems `mdos` will be replaced with `gpt`, but the steps will be the same.



Notice when we ran `ls (hd0,1)/boot/` we can see the two files we need! Both `vmlinuz-5.15.0-kali3-amd64` and `initrd.img-5.15.0-kali3-amd64` exist in the `(hd0,1)/boot/` directory.

You may have noticed that the `ls /` command output the files `vmlinuz` and `initrd.img` - that's because some systems create symlinks to the last used kernel and initrd images in the root directory of the filesystem. You can use these if you're sure they're what you want, or you can just use the full path to the files under the `/boot/` directory.

We continue by setting the root filesystem to the `(hd0,1)` device with the following commands

```
grub> set root=(hd0,1)
```

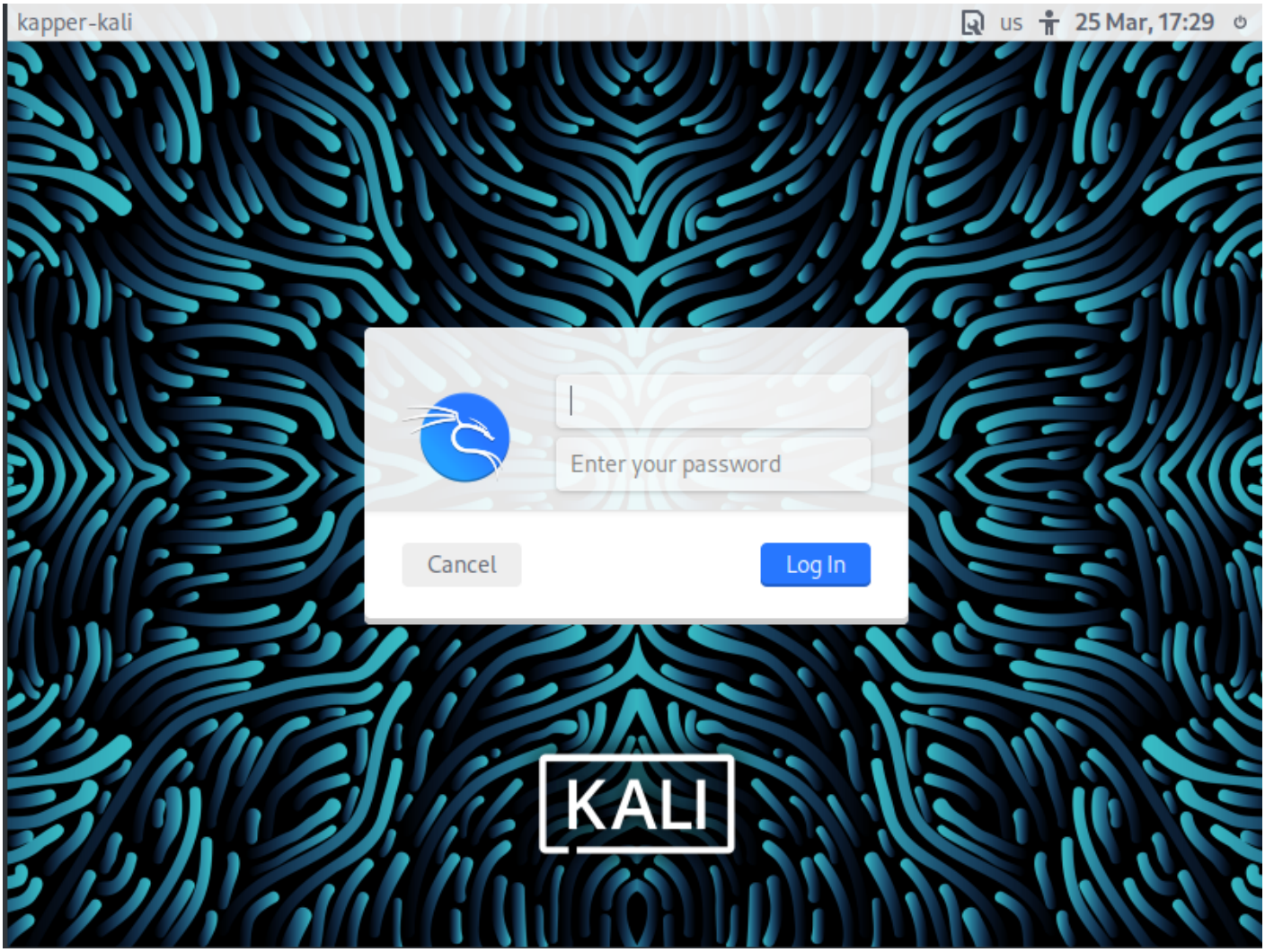
Now you'll notice that running `ls /boot/` points us to the directory with the files we need, and at this point we are almost ready to boot the system.

Next, we run the following commands to set the linux kernel and `initrd` image, then we boot the system by running the `boot` command. The system will boot normally and you'll be asked to login.

```
grub> linux /boot/vmlinuz-5.15.0-kali3-amd64 root=/dev/sda1
grub> initrd /boot/initrd.img-5.15.0-kali3-amd64
grub> boot
```

```
         Starting Create Static Device Nodes in /dev...
[  OK  ] Finished Flush Journal to Persistent Storage.
[  OK  ] Finished Create Static Device Nodes in /dev.
[  OK  ] Reached target Preparation for Local File Systems.
[  OK  ] Reached target Local File Systems.
         Starting Set console font and keymap...
         Starting Raise network interfaces...
         Starting Preprocess NFS configuration...
         Starting Tell Plymouth To Write Out Runtime Data...
         Starting Create Volatile Files and Directories...
         Starting Rule-based Manager for Device Events and Files...
[  OK  ] Finished Tell Plymouth To Write Out Runtime Data.
[  OK  ] Finished Preprocess NFS configuration.
[  OK  ] Finished Set console font and keymap.
[  OK  ] Reached target NFS client services.
[  OK  ] Reached target Preparation for Remote File Systems.
[  OK  ] Reached target Remote File Systems.
[  OK  ] Finished Create Volatile Files and Directories.
[  OK  ] Started Entropy Daemon based on the HAVEGE algorithm.
         Starting Record System Boot/Shutdown in UTMP...
[  OK  ] Finished Record System Boot/Shutdown in UTMP.
[  OK  ] Started Rule-based Manager for Device Events and Files.
         Starting Show Plymouth Boot Screen...
[    2.545353] ACPI: AC: AC Adapter [AC] (on-line)
[  OK  ] Started Show Plymouth Boot Screen.
[  OK  ] Started Forward Password Requests to Plymouth Directory Watch.
[  OK  ] Reached target Local Encrypted Volumes.
[  OK  ] Reached target Path Units.
[    2.567532] sd 0:0:0:0: Attached scsi generic sg0 type 0
[    2.569340] sr 2:0:0:0: Attached scsi generic sg1 type 5
[    2.579522] vboxguest: host-version: 6.1.32r149290 0x8000000f
[    2.583395] vbg_heartbeat_init: Setting up heartbeat to trigger every 2000 milliseconds
[    2.583734] input: VirtualBox mouse integration as /devices/pci0000:00/0000:00:04.0/input/input6
[    2.606161] vboxguest: misc device minor 125, IRQ 20, I/O port d040, MMIO at 0x00000000f0400000 (
size 0x0000000000400000)
[    2.610515] input: PC Speaker as /devices/platform/pcspkr/input/input7

_
```



Linux - Rescue GRUB

Revision #3
Created 25 January 2022 15:49:58 by Shaun Reed
Updated 25 March 2022 22:10:59 by Shaun Reed