

00. Introduction to Kali Linux

What is Kali Linux ?

[Kali Linux](#) is an advanced Penetration Testing and Security Auditing Linux distribution.

Kali Linux Features

Kali is a complete re-build of [BackTrack Linux](#), adhering completely to [Debian](#) development standards. All-new infrastructure has been put in place, all tools were reviewed and packaged, and we use [Git](#) for our VCS.

- **More than 300 penetration testing tools:** After reviewing every tool that was included in BackTrack, we eliminated a great number of tools that either did not work or had other tools available that provided similar functionality.
- **Free and always will be:** Kali Linux, like its predecessor, is completely free and always will be. You will never, ever have to pay for Kali Linux.
- **Open source Git tree:** We are huge proponents of open source software and our [development tree](#) is available for all to see and all sources are available for those who wish to tweak and rebuild packages.
- **FHS compliant:** Kali has been developed to adhere to the [Filesystem Hierarchy Standard](#), allowing all Linux users to easily locate binaries, support files, libraries, etc.
- **Vast wireless device support:** We have built Kali Linux to support as many wireless devices as we possibly can, allowing it to run properly on a wide variety of hardware and making it compatible with numerous USB and other wireless devices.
- **Custom kernel patched for injection:** As penetration testers, the development team often needs to do wireless assessments so our kernel has the latest injection patches included.
- **Secure development environment:** The Kali Linux team is made up of a small group of trusted individuals who can only commit packages and interact with the repositories while using multiple secure protocols.
- **GPG signed packages and repos:** All Kali packages are signed by each individual developer when they are built and committed and the repositories subsequently sign the packages as well.
- **Multi-language:** Although pentesting tools tend to be written in English, we have ensured that Kali has true multilingual support, allowing more users to operate in their native language and locate the tools they need for the job.
- **Completely customizable:** We completely understand that not everyone will agree with our design decisions so we have made it as easy as possible for our more adventurous users to [customize Kali Linux](#) to their liking, all the way down to the kernel.
- **ARMEL and ARMHF support:** Since ARM-based systems are becoming more and more prevalent and inexpensive, we knew that [Kali's ARM support](#) would need to be as robust as we could manage, resulting in working installations for both [ARMEL and ARMHF](#) systems. Kali Linux has ARM repositories integrated

with the mainline distribution so tools for ARM will be updated in conjunction with the rest of the distribution. Kali is currently available for the following ARM devices:

- [rk3306 mk/ss808](#)
- [Raspberry Pi](#)
- [ODROID U2/X2](#)
- [Samsung Chromebook](#)
- [EfikaMX](#)
- [Beaglebone Black](#)
- [CuBox](#)
- [Galaxy Note 10.1](#)

Kali is specifically tailored to penetration testing and therefore, all documentation on this site assumes prior knowledge of the Linux operating system.

Should I Use Kali Linux?

Differences Between Kali Linux and Debian

Kali Linux is geared towards professional penetration testing and security auditing. As such, several core changes have been implemented in Kali Linux which reflect these needs:

1. **Single user, root access by design:** Due to the nature of security audits, Kali linux is designed to be used in a "[single, root user](#)" scenario.
2. **Network services disabled by default:** Kali Linux contains sysvinit hooks which [disable network services](#) by default. These hooks allow us to install various services on Kali Linux, while ensuring that our distribution remains secure by default, no matter what packages are installed. Additional services such as Bluetooth are also blacklisted by default.
3. **Custom Linux kernel:** Kali Linux uses an upstream kernel, patched for wireless injection.

Is Kali Linux Right For You?

As the distribution developers, one would likely expect us to recommend that everyone use Kali Linux. The fact of the matter is however, that Kali is a Linux distribution specifically geared towards professional penetration testing and security auditing and as such, it is **NOT** a recommended distribution for those unfamiliar with Linux.

In addition, misuse of security tools within your network, particularly without permission, may cause irreparable damage and result in significant consequences.

If you are looking for a Linux distribution to learn the basics of Linux and need a good starting point, Kali Linux is not the ideal distribution for you. You may want to begin with [Ubuntu](#) or [Debian](#) instead.

Kali Linux Default Passwords

Kali Linux Default root Password is toor

Default root Password

During installation, Kali Linux allows users to configure a password for the *root* user. However, should you decide to boot the live image instead, the i386, amd64, VMWare and ARM images are configured with the **default root password** - **“toor”**, without the quotes.

01. Downloading Kali Linux

Download Official Kali Images

Alert! Always make certain you are downloading Kali Linux from official sources and be sure to verify the SHA1 checksums against our official values. It would be easy for a malicious entity to modify a Kali installation to contain malicious code and host it unofficially.

Official Kali Linux Images

ISO Files

Kali Linux is available as a bootable ISO in both 32 and 64-bit formats.

- [Download Kali ISOs](#)

VMware Images

Kali is available as a pre-made VMware virtual machine with VMware Tools installed. The VMware image is available in a 32-bit PAE format.

- [Download Kali VMware Images](#)

ARM Images

Due to the nature of the ARM architecture, it is not possible to have a single image that will work across all ARM devices. We have [Kali Linux ARM images](#) available for the following devices:

- rk3306 mk/ss808
- Raspberry Pi
- ODROID-U2/X2
- MK802/MK802 II
- Samsung Chromebook

Verifying SHA1 Checksums of Downloaded Images

When you download an image, be sure to download the SHA1SUMS and SHA1SUMS.gpg files that are next to

the downloaded image (i.e. in the same directory on the server).

Ensure the Origin of the SHA1SUMS File

Before verifying the checksums of the image, you must ensure that the SHA1SUMS file is the one generated by Kali. That's why the file is signed by Kali's official key with a detached signature in SHA1SUMS.gpg. Kali's official key can be downloaded in one of two ways:

```
$ wget -q -O - http://archive.kali.org/archive-key.asc | gpg --import
# or
$ gpg --keyserver subkeys.pgp.net --recv-key 44C6513A8E4FB3D30875F758ED444FF07D8D0BF6
```

Once you have downloaded both SHA1SUMS and SHA1SUMS.gpg, you can verify the signature as follows:

```
$ gpg --verify SHA1SUMS.gpg SHA1SUMS
gpg: Signature made Thu Mar  7 21:26:40 2013 CET using RSA key ID 7D8D0BF6
gpg: Good signature from "Kali Linux Repository <devel@kali.org>"
```

If you don't get that "Good signature" message or if the key ID doesn't match, then you should stop the process and review whether you downloaded the images from a legitimate Kali mirror. If the SHA1SUMS file is the one provided by Kali, then you can verify that the image downloaded has the required checksum. You can either generate the checksum and do a manual comparison with what's listed in SHA1SUMS or use a tool that knows how to verify those checksums. **TODO: explain how to use GPG on OS X and Windows. See <https://www.torproject.org/docs/verifying-signatures.html.en> for inspiration.**

Verifying SHA1 Checksums on Linux

With a manual comparison:

```
$ sha1sum kali-linux-1.0-i386.iso
796e32f51d1bf51e838499c326c71a1c952cc052 kali-linux-1.0-i386.iso
$ grep kali-linux-1.0-i386.iso SHA1SUMS
796e32f51d1bf51e838499c326c71a1c952cc052 kali-linux-1.0-i386.iso
```

By using sha1sum -c:

```
grep kali-linux-1.0-i386.iso SHA1SUMS | sha1sum -c  
kali-linux-1.0-i386.iso: OK
```

Verifying SHA1 Checksums on OSX

With a manual comparison:

```
$ shasum kali-linux-1.0-i386.iso  
796e32f51d1bf51e838499c326c71a1c952cc052 kali-linux-1.0-i386.iso  
$ grep kali-linux-1.0-i386.iso SHA1SUMS  
796e32f51d1bf51e838499c326c71a1c952cc052 kali-linux-1.0-i386.iso
```

Verifying SHA1 Checksums on Windows

Windows does not have the native ability to calculate SHA1 checksums so you will need a utility such as [Free MD5 SHA1 verifier](#) to verify your download.

Generate an Updated Kali ISO

Kali Linux allows you to generate updated ISOs of Kali using Debian [live-build](#) scripts on the fly. The easiest way to generate these images is from within a Kali Linux environment as follows.

You will first need to install the *live-build* and *cdebootstrap* packages:

```
apt-get install git live-build cdebootstrap
```

Next, we clone the Kali *cdimage* Git repository as follows:

```
git clone git://git.kali.org/live-build-config.git
```

Now you can change to the *live* directory under *cdimage.kali.org* and build your ISO.

```
cd live-build-config  
lb clean --purge  
lb config  
lb build
```

The live build scripts allow for complete customization of Kali Linux images. For more information about Kali live build scripts, check out our [Kali customization page](#).

02. Building Custom Kali Images

Live Build a Custom Kali ISO

Build Your Own Kali ISO - Introduction

Building a customized Kali ISO is easy, fun, and rewarding. You can configure virtually every aspect of your custom Kali ISO build using the Debian [live-build](#) scripts. These scripts allow one to easily build live system images by providing a framework that uses a configuration set to automate and customize all aspects of building the image. We have adopted these scripts and use them for the official Kali ISO releases.

Prerequisites

Ideally, you should build your custom Kali ISO from within a pre-existing Kali environment. However, if this is not the case for you, make sure you are using the latest version of live-build (in the 3.x branch which targets Debian wheezy).

Getting Ready

We first need to prepare the Kali ISO build environment with the following commands:

```
apt-get install git live-build cdebootstrap kali-archive-keyring
git clone git://git.kali.org/live-build-config.git
cd live-build-config
lb config
```

Configuring the Kali ISO Build (Optional)

Through the **config** directory, your ISO build supports significant customization options, which are well documented on the Debian [live build 3.x](#) page. However, for the impatient, the following configuration files are of particular interest:

config/package-lists/kali.list.chroot - contains the list of packages to install in the Kali ISO. You can choose specific packages to be installed, while dropping others. This is also where you can [change your Kali ISO Desktop Environment](#) (KDE, Gnome, XFCE, LXDE, etc).

hooks/ - The hooks directory allows us to hook scripts in various stages of the Kali ISO live build. For more information about hooks, refer to the [live build manual](#). As an example, Kali adds its forensic menu this way:

```
$ cat config/hooks/forensic-menu.binary
#!/bin/sh

cat >>binary/isolinux/live.cfg <<END

label live-forensic
  menu label ^Live (forensic mode)
  linux /live/vmlinuz
  initrd /live/initrd.img
  append boot=live noconfig username=root hostname=kali noswap noautomount
END
```

Building the ISO

Before you generate your ISO, you can specify your required architecture, choosing either amd64 or i386. Also note that “lb build” requires root rights. If you do not specify an architecture, live build will generate an ISO with the same architecture as the host machine.

If you want to build a 64 bit ISO on a 32 bit Kali system, make sure you enable multi archi support:

```
dpkg --add-architecture amd64
apt-get update
```

Configure live-build to generate with a 64 bit or 32 bit ISO:

```
lb config --architecture amd64 # for 64 bit
# ...or...
```

```
lb config --architecture i386 # for 32 bit

lb build
```

The last command will take a while to complete, as it downloads all of the required packages needed to create your ISO. Good time for a coffee.

Building Kali Linux for older i386 architecture

The Kali Linux i386 ISO has PAE enabled. If you require a default kernel for older hardware, you need to rebuild a Kali Linux ISO. The rebuilding process is much the same as above, other than the **686-pae** parameter that needs to be changed to **486** in **auto/config** :

```
apt-get install git live-build cdebootstrap kali-archive-keyring
git clone git://git.kali.org/live-build-config.git
cd live-build-config
sed -i 's/686-pae/486/g' auto/config
lb clean
lb config --architecture i386
lb build
```

Speeding up future builds

If you plan to build custom ISOS often, you might want to cache kali packages locally for future builds. This can easily be done by installing **apt-cacher-ng**, and configuring the *http_proxy* environment variable before every build.

```
apt-get install apt-cacher-ng
/etc/init.d/apt-cacher-ng start
export http_proxy=http://localhost:3142/
.... # setup and configure your live build
```

lb build

Customize the Kali Desktop

Changing the Kali Desktop Environment

Although Kali Linux uses Gnome for its default desktop environment, we recognize that not all users wish to use Gnome so we have made it simple to change to a WM of your choosing. To build your own Kali ISO image with a custom Desktop Environment, start by following the [Live Build a Custom Kali ISO guide](#). Before building your ISO, edit the last section of **config/package-lists/kali.list.chroot** to contain the entries related to the desktop environment of your choice. The section starts with this comment:

```
# Graphical desktops depending on the architecture
#
# You can replace all the remaining lines with a list of the
# packages required to install your preferred graphical desktop
# or you can just comment everything except the packages of your
# preferred desktop.
```

- [KDE](#)
- [Gnome](#)
- [LXDE](#)
- [XFCE](#)
- [i3WM](#)
- [MATE](#)

```
kali-defaults
kali-root-login
desktop-base
kde-plasma-desktop
```

```
gnome-core  
kali-defaults  
kali-root-login  
desktop-base
```

```
kali-defaults  
kali-root-login  
desktop-base  
lxde
```

```
kali-defaults  
kali-root-login  
desktop-base  
xfce4  
xfce4-places-plugin
```

```
# cheers to 0xerror  
xorg  
dmenu  
conky  
i3
```

The “MATE” desktop is not included by default in our repositories, and requires a few more steps to integrate into a Kali build.

```
echo "deb http://repo.mate-desktop.org/debian wheezy main" >> /etc/apt/sources.list
apt-get update
apt-get install mate-archive-keyring
```

```
# apt-get install git live-build cdebootstrap
# git clone git://git.kali.org/live-build-config.git
cd live-build-config
mkdir config/archives
echo "deb http://repo.mate-desktop.org/debian wheezy main" > config/archives/mate.list.binary
echo "deb http://repo.mate-desktop.org/debian wheezy main" > config/archives/mate.list.chroot
cp /usr/share/keyrings/mate-archive-keyring.gpg config/archives/mate.key.binary
cp /usr/share/keyrings/mate-archive-keyring.gpg config/archives/mate.key.chroot
echo "sleep 20" >> config/hooks/z_sleep.chroot
```

```
# add mate desktop to the packages list:
nano config/package-lists/kali.list.chroot
```

```
# after editing, it should look like this:
xorg
mate-archive-keyring
mate-core
mate-desktop-environment
```


03. Installing Kali Linux

Kali Linux Hard Disk Install

Kali Linux Installation Requirements

Installing Kali Linux on your computer is an easy process. First, you'll need compatible computer hardware. Kali is supported on i386, amd64, and ARM (both armel and armhf) platforms. The hardware requirements are minimal as listed below, although better hardware will naturally provide better performance. The i386 images have a default [PAE](#) kernel, so you can run them on systems with over 4GB of RAM. [Download Kali Linux](#) and either burn the ISO to DVD, or [prepare a USB stick with Kali Linux Live](#) as the installation medium. If you do not have a DVD drive or USB port on your computer, check out the [Kali Linux Network Install](#).

Installation Prerequisites

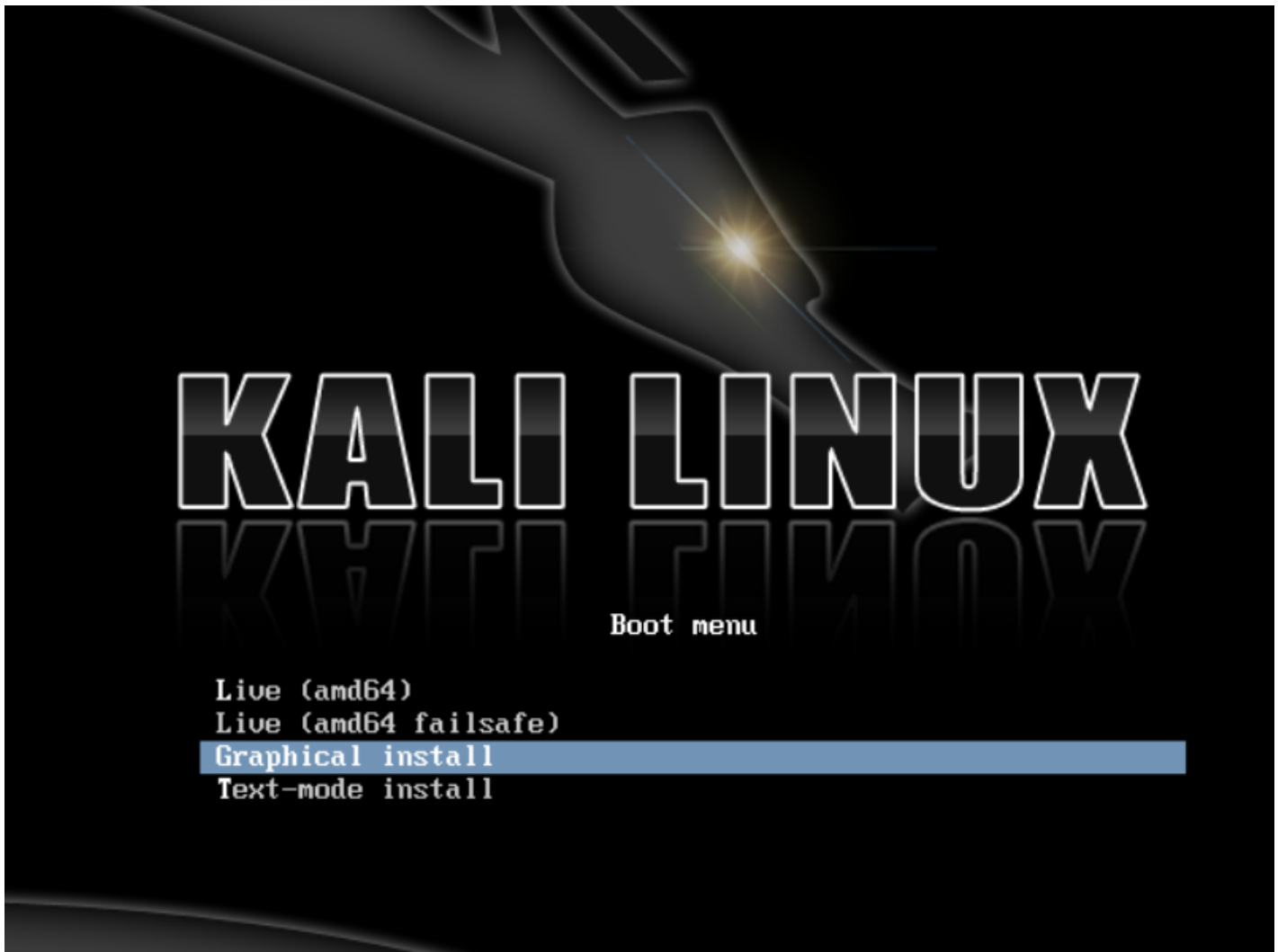
- A minimum of 8 GB disk space for the Kali Linux install.
- For i386 and amd64 architectures, a minimum of 512MB RAM.
- CD-DVD Drive / USB boot support

Preparing for the Installation

1. [Download Kali linux](#).
2. Burn The Kali Linux ISO to DVD or [Image Kali Linux Live to USB](#).
3. Ensure that your computer is set to boot from CD / USB in your BIOS.

Kali Linux Installation Procedure

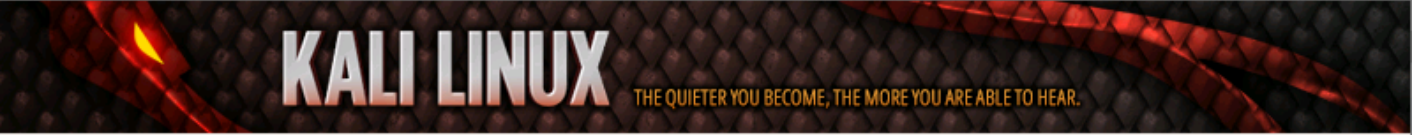
1. To start your installation, boot with your chosen installation medium. You should be greeted with the Kali Boot screen. Choose either *Graphical* or *Text-Mode* install. In this example, we chose a GUI install.



2. Select your preferred language and then your country location. You'll also be prompted to configure your keyboard with the appropriate keymap.



- The installer will copy the image to your hard disk, probe your network interfaces, and then prompt you to enter a hostname for your system. In the example below, we've entered "kali" as our hostname.



Configure the network

Please enter the hostname for this system.

The hostname is a single word that identifies your system to the network. If you don't know what your hostname should be, consult your network administrator. If you are setting up your own home network, you can make something up here.

Hostname:

[Screenshot](#) [Go Back](#) [Continue](#)

4. Enter a robust password for the root account.



Set up users and passwords

You need to set a password for 'root', the system administrative account. A malicious or unqualified user with root access can have disastrous results, so you should take care to choose a root password that is not easy to guess. It should not be a word found in dictionaries, or a word that could be easily associated with you.

A good password will contain a mixture of letters, numbers and punctuation and should be changed at regular intervals.

The root user should not have an empty password. If you leave this empty, the root account will be disabled and the system's initial user account will be given the power to become root using the "sudo" command.

Note that you will not be able to see the password as you type it.

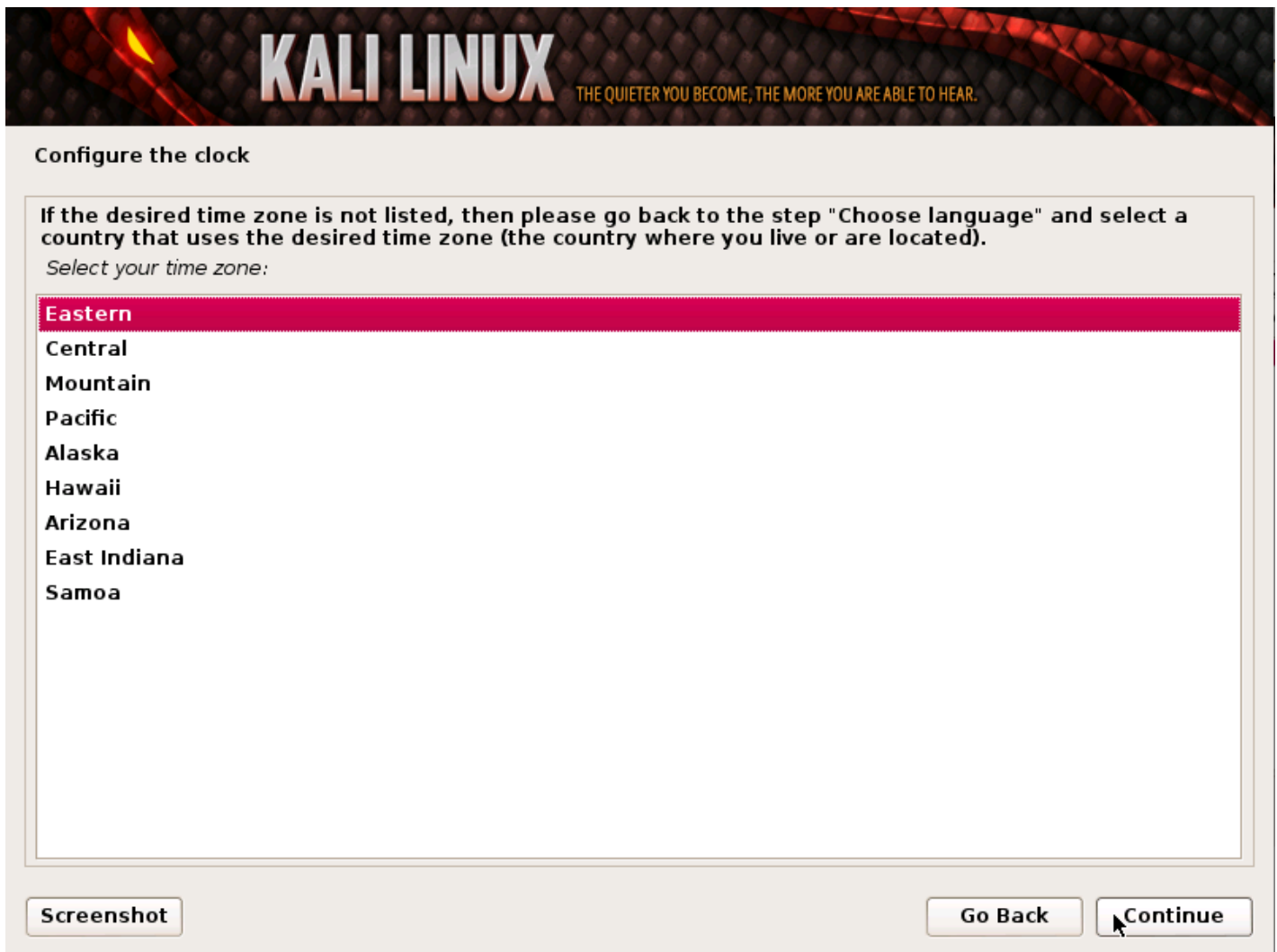
Root password:

Please enter the same root password again to verify that you have typed it correctly.

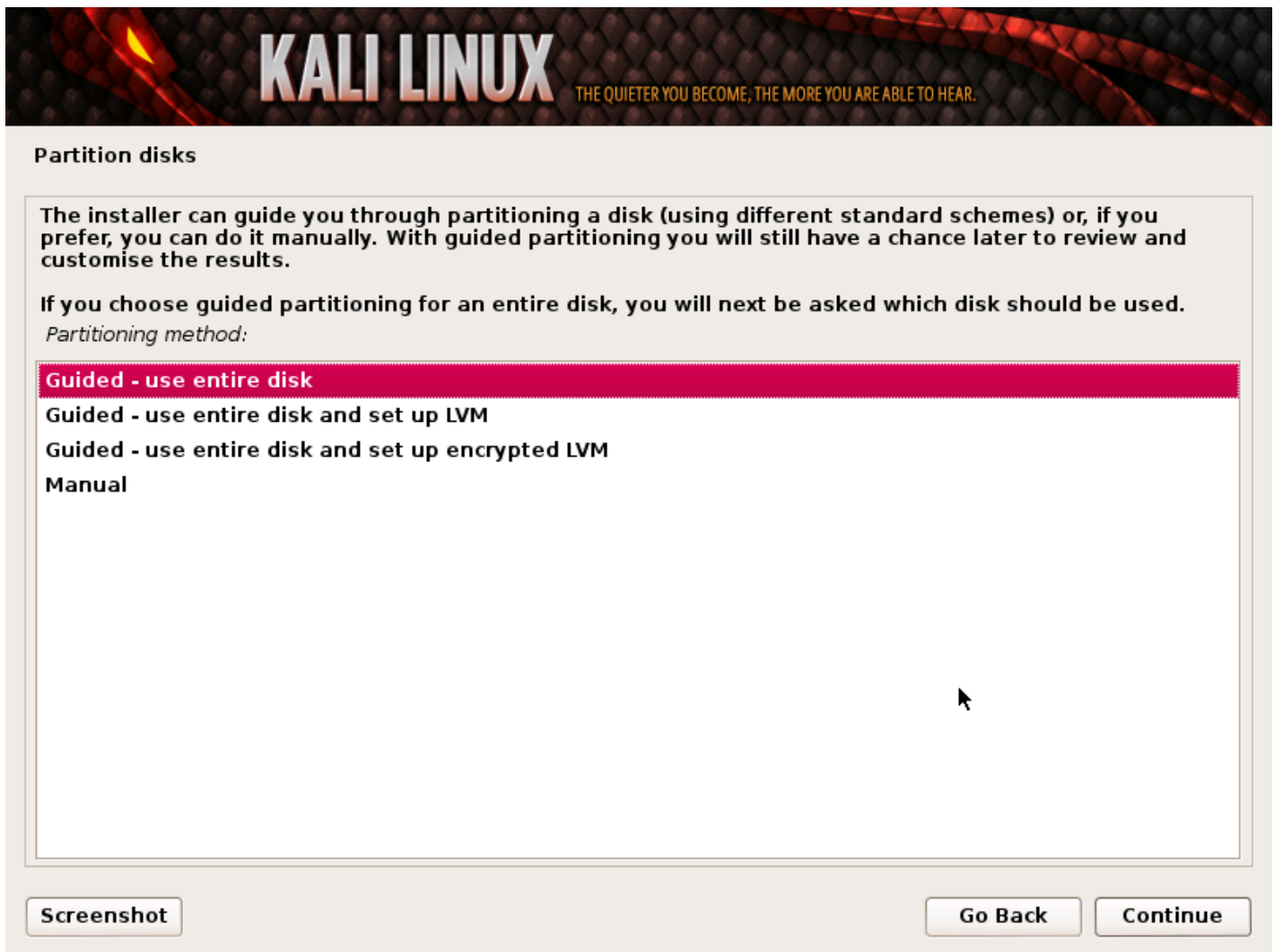
Re-enter password to verify:

[Screenshot](#) [Go Back](#) [Continue](#)

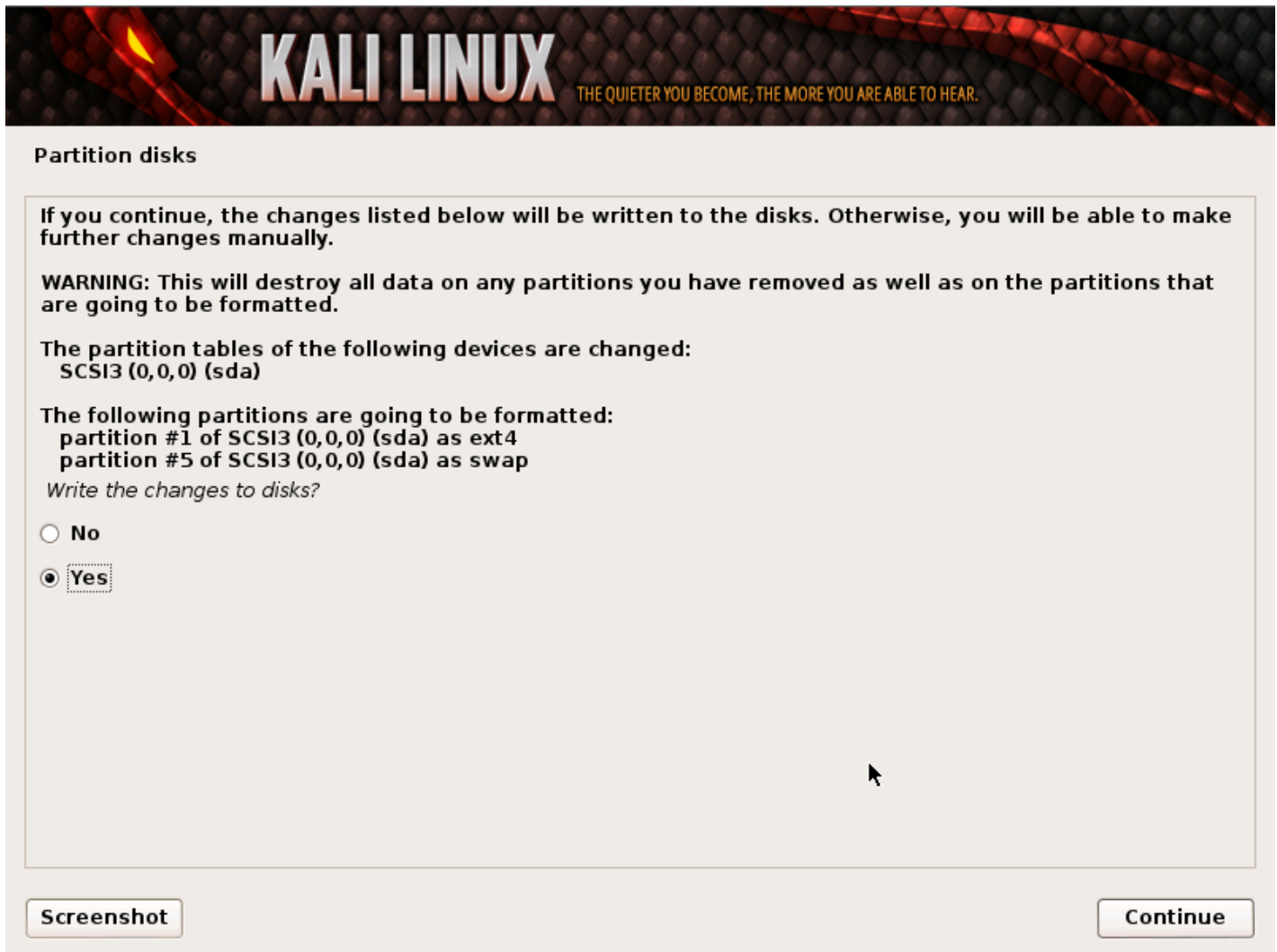
5. Next, set your time zone.



- The installer will now probe your disks and offer you four choices. In our example, we're using the entire disk on our computer and not configuring LVM (logical volume manager). Experienced users can use the "Manual" partitioning method for more granular configuration options.

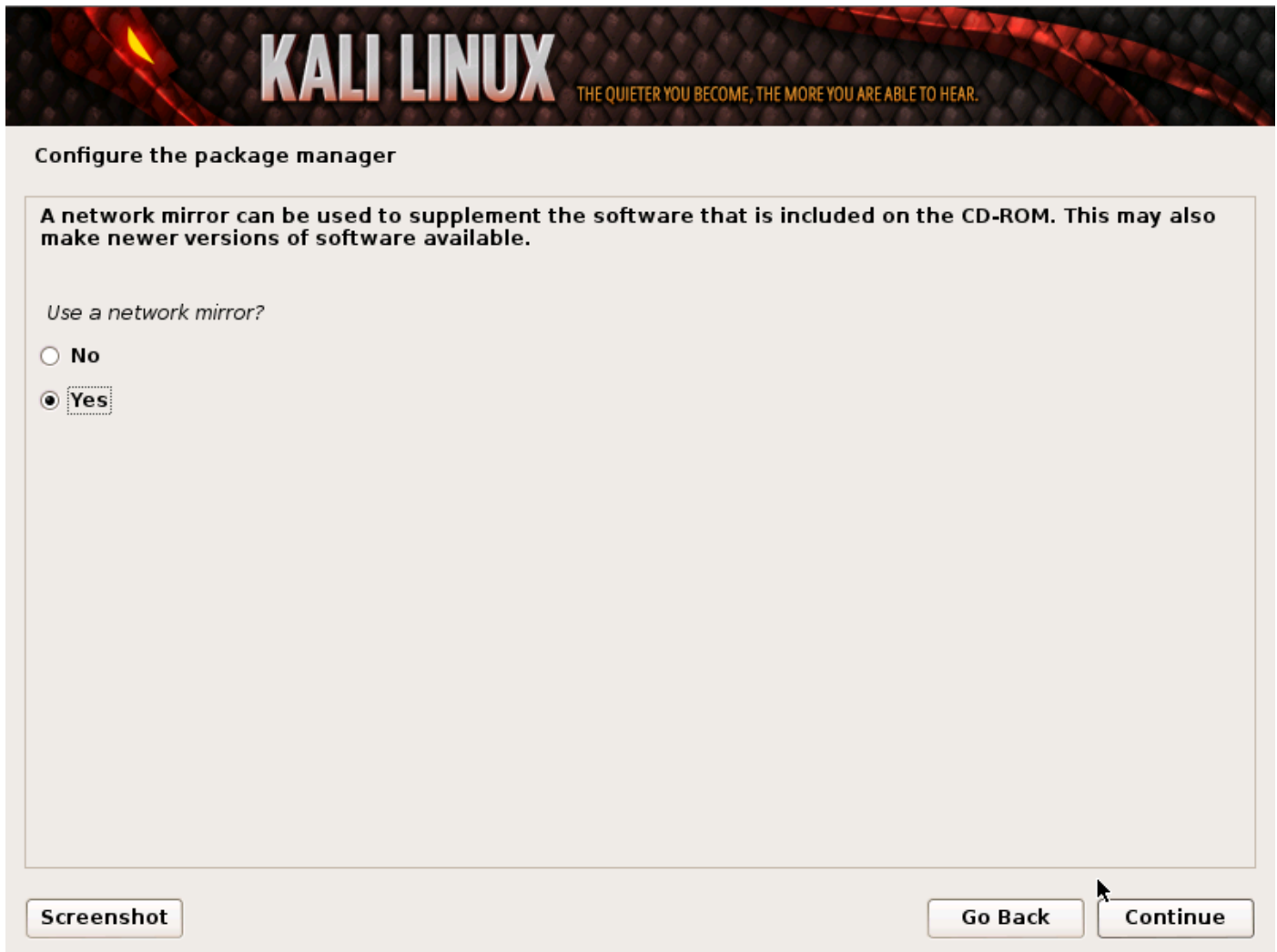


- Next, you'll have one last chance to review your disk configuration before the installer makes irreversible changes. After you click *Continue*, the installer will go to work and you'll have an almost finished installation.



8. Configure network mirrors. Kali uses a central repository to distribute applications. You'll need to enter any appropriate proxy information as needed.

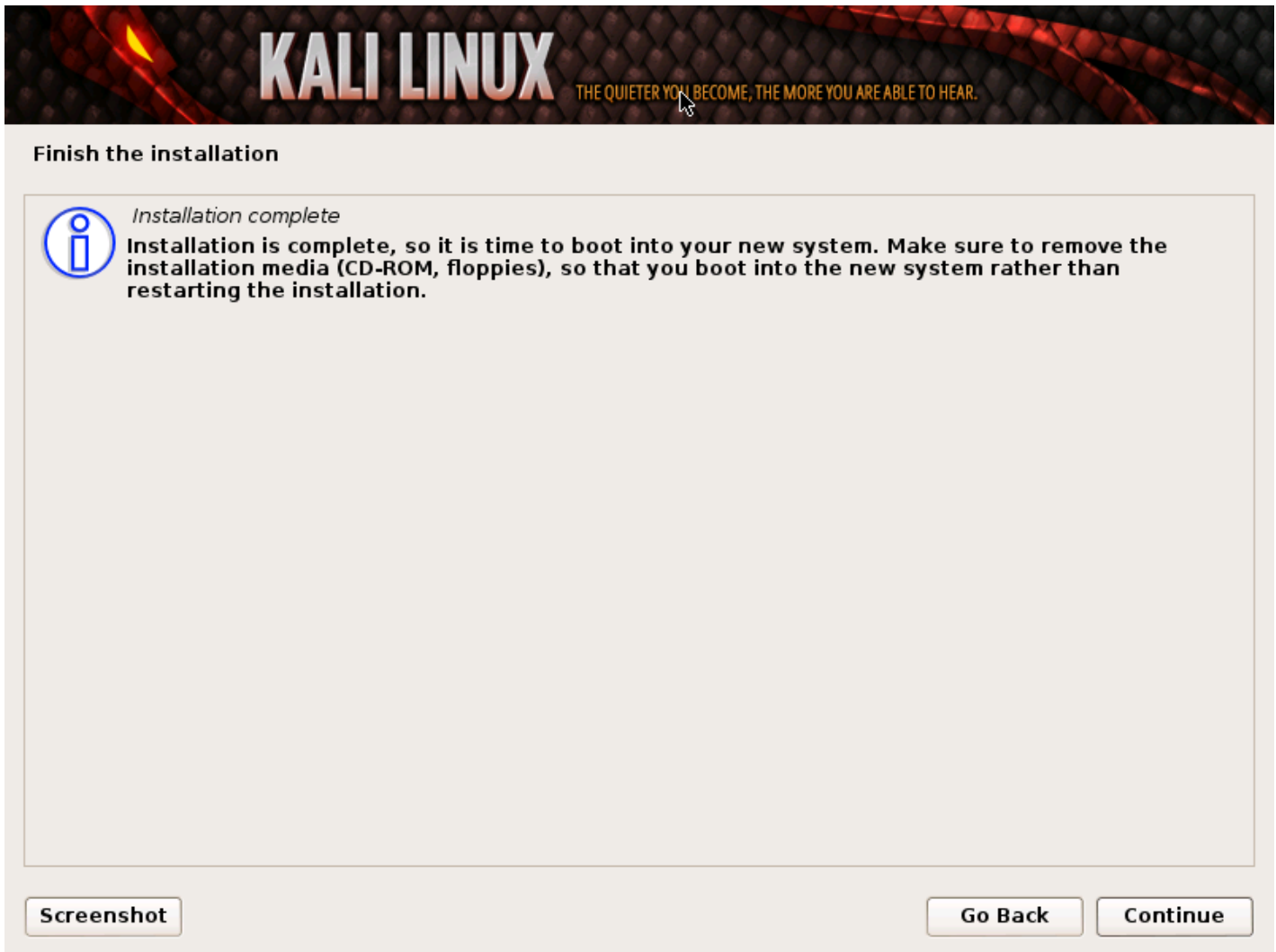
NOTE! If you select "NO" in this screen, you will **NOT** be able to install packages from Kali repositories.



9. Next, install GRUB.



10. Finally, click Continue to reboot into your new Kali installation.



Post Installation

Now that you've completed installing Kali Linux, it's time to customize your system. The [Kali General Use](#) section of our site has more information and you can also find tips on how to get the most out of Kali in our [User Forums](#).

Dual Boot Kali with Windows

Kali Linux Dual Boot with Windows

Installing Kali alongside a Windows installation can be quite useful. However, you need to exercise caution during the setup process. First, make sure that you've backed up any important data on your Windows installation. Since you'll be modifying your hard drive, you'll want to store this backup on external media. Once you've completed the backup, we recommend you peruse [Kali Linux Hard Disk Install](#), which explains the normal procedure for a basic Kali install.

In our example, we will be installing Kali Linux alongside an installation of Windows 7, which is currently taking up 100% of the disk space in our computer. We will start by resizing our current Windows partition to occupy less space and then proceed to install Kali Linux in the newly-created empty partition.

[Download Kali Linux](#) and either burn the ISO to DVD, or [prepare a USB stick with Kali linux Live](#) as the installation medium. If you do not have a DVD or USB port on your computer, check out the [Kali Linux Network Install](#). Ensure you have:

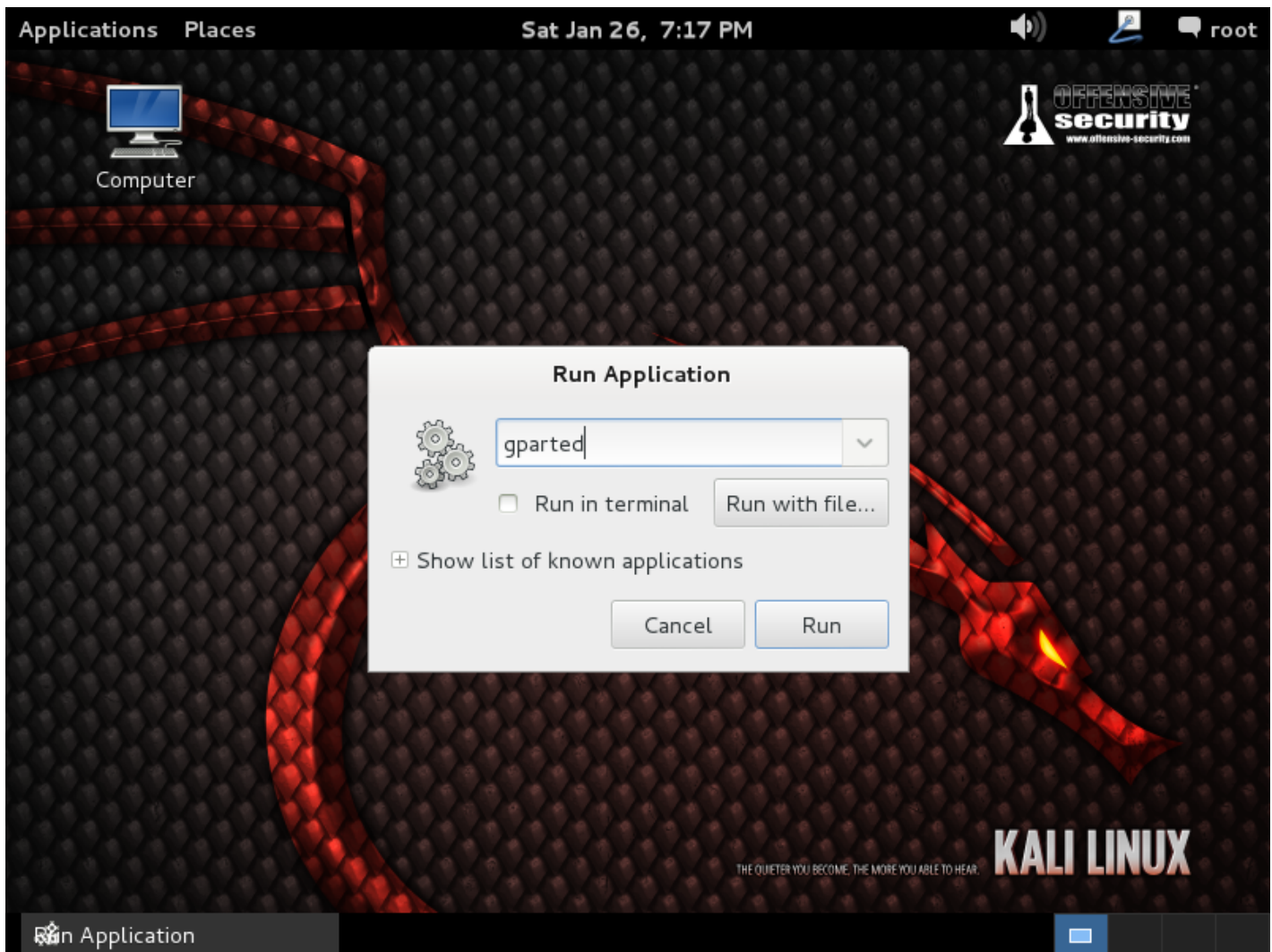
- Minimum of 8 GB free disk space on Windows
- CD-DVD / USB boot support

Preparing for the Installation

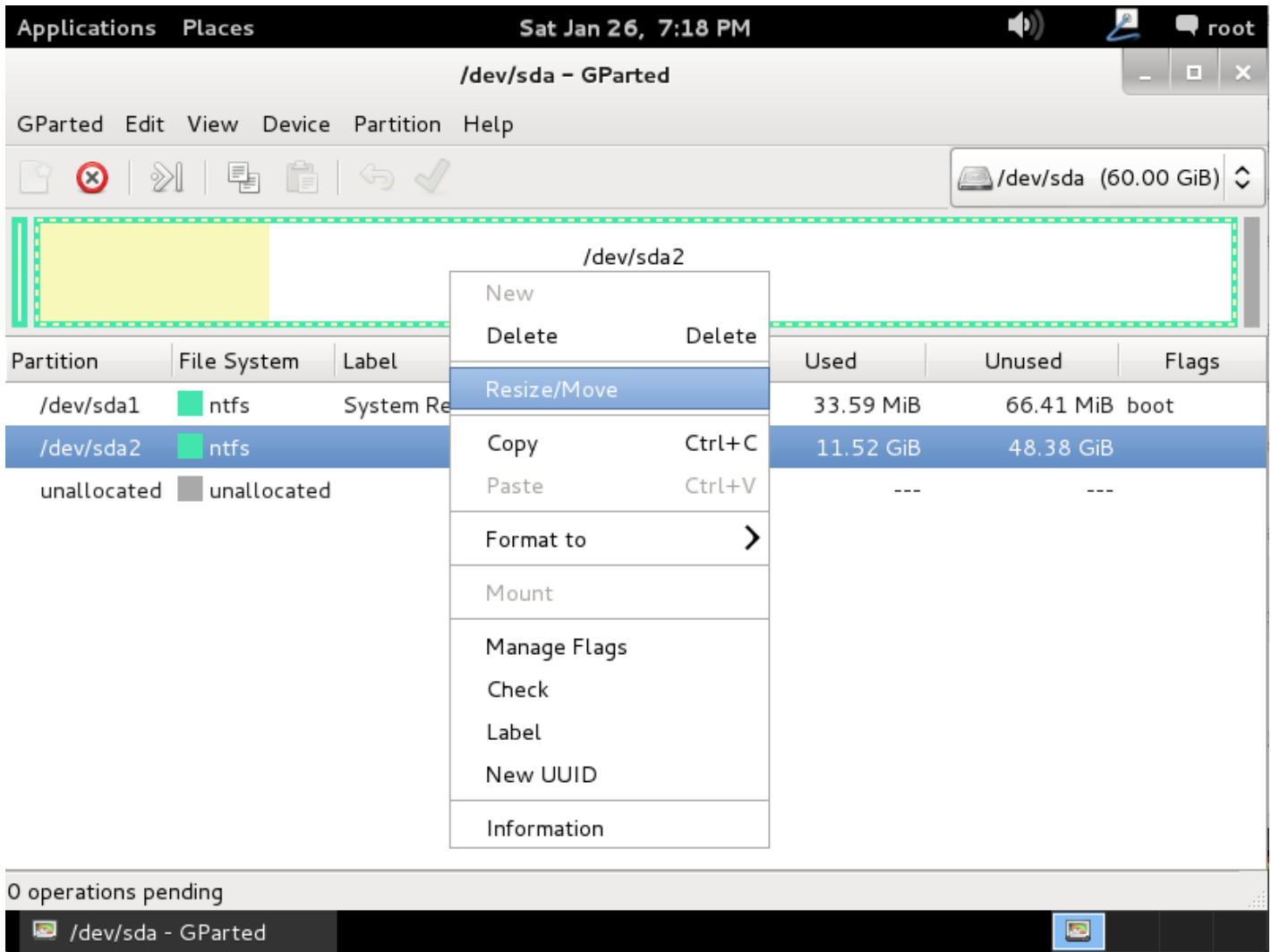
1. [Download Kali Linux](#).
2. Burn The Kali Linux ISO to DVD or [copy Kali Linux Live to USB](#).
3. Ensure that your computer is set to boot from CD / USB in your BIOS.

Dual Boot Installation Procedure

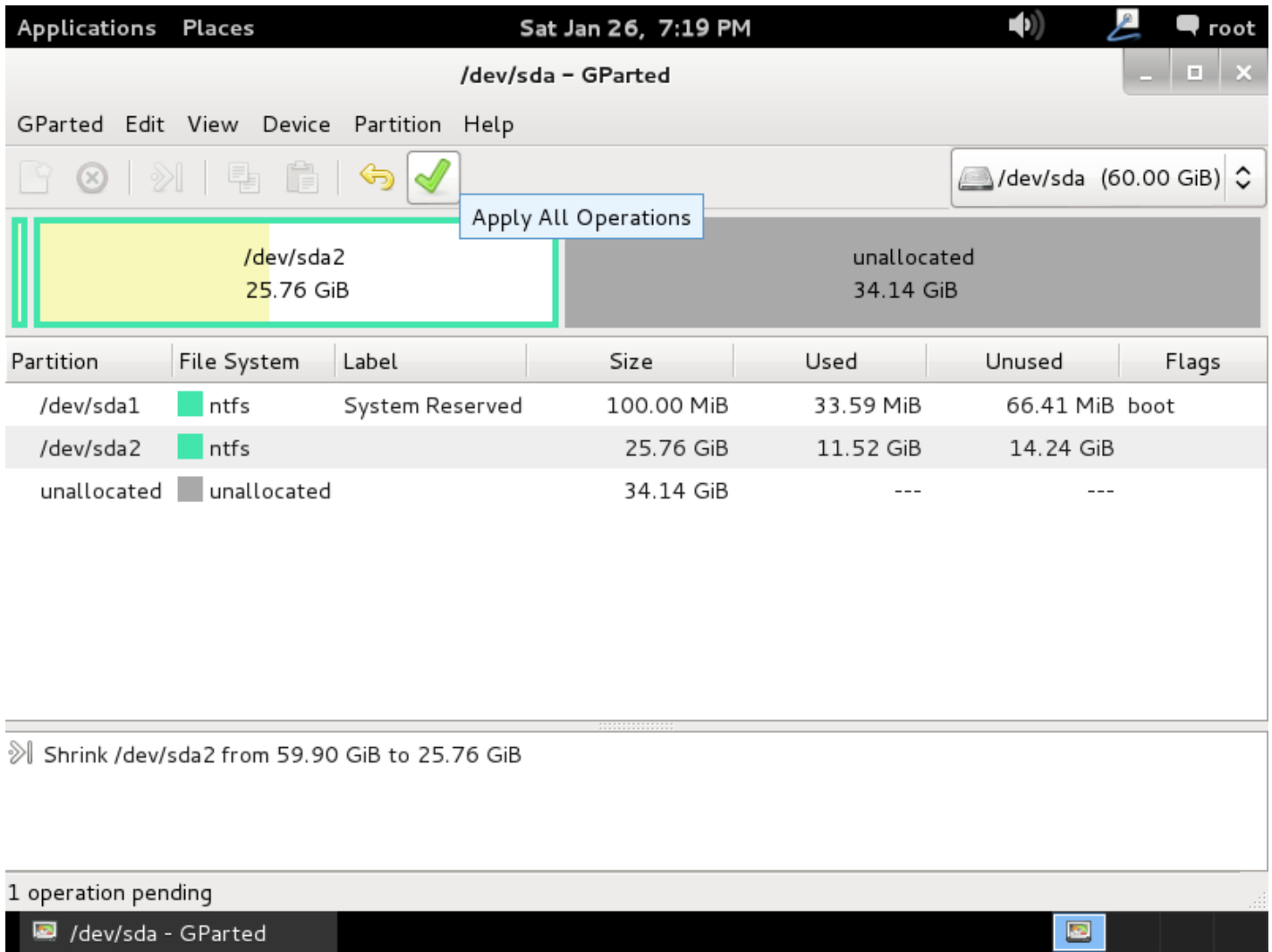
1. To start your installation, boot with your chosen installation medium. You should be greeted with the Kali Boot screen. Select *Live*, and you should be booted into the Kali Linux default desktop.
2. Now launch the **gparted** program. We'll use **gparted** to shrink the existing Windows partition to give us enough room to install Kali Linux.



3. Select your Windows partition. Depending on your system, it will usually be the second, larger partition. In our example, there are two partitions; the first is the System Recovery partition, and Windows is actually installed in `/dev/sda2`. Resize your Windows partition and leave enough space (8GB minimum) for the Kali installation.

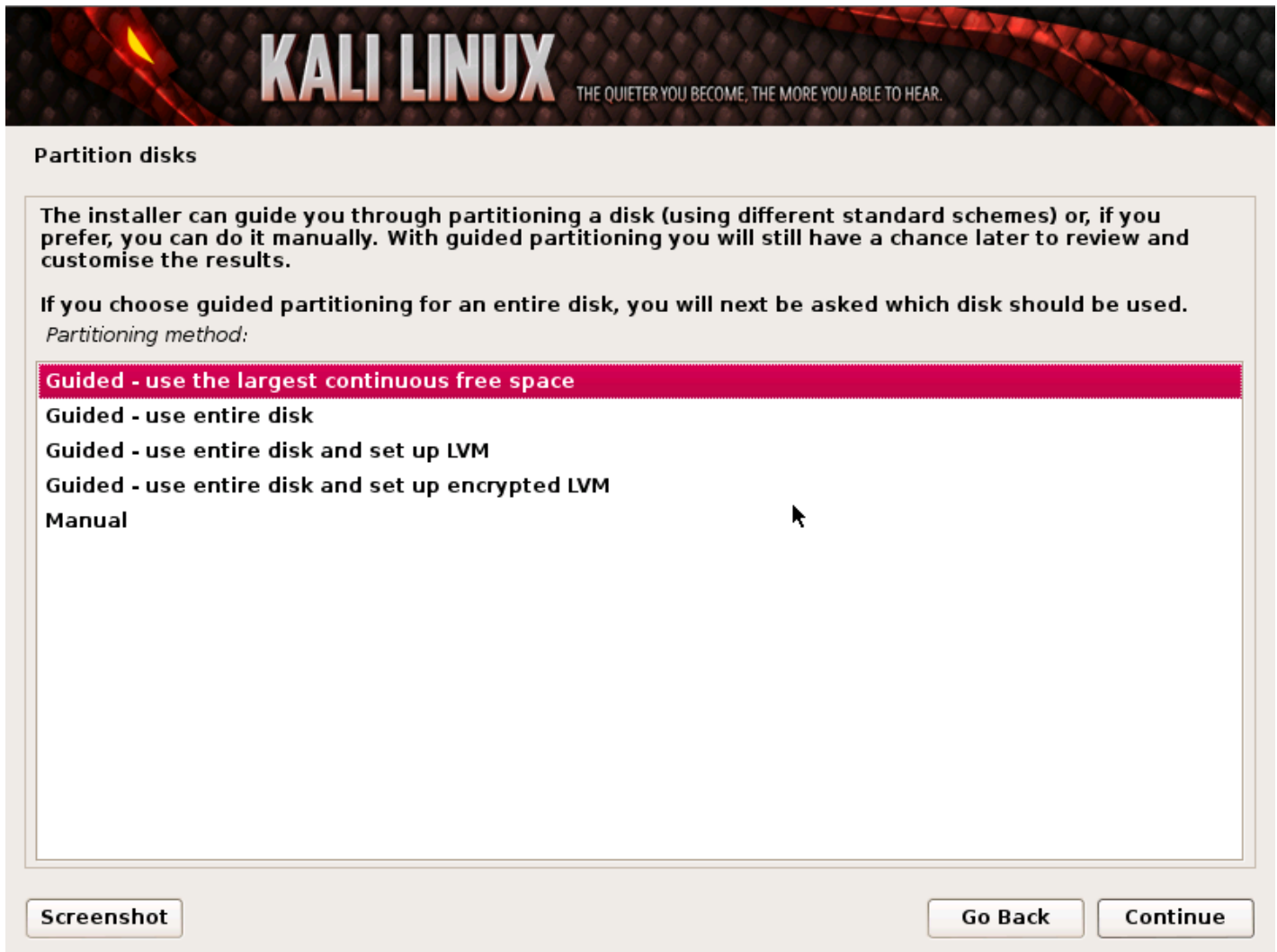


- Once you have resized your Windows partition, ensure you “Apply All Operations” on the hard disk. Exit **gparted** and reboot.

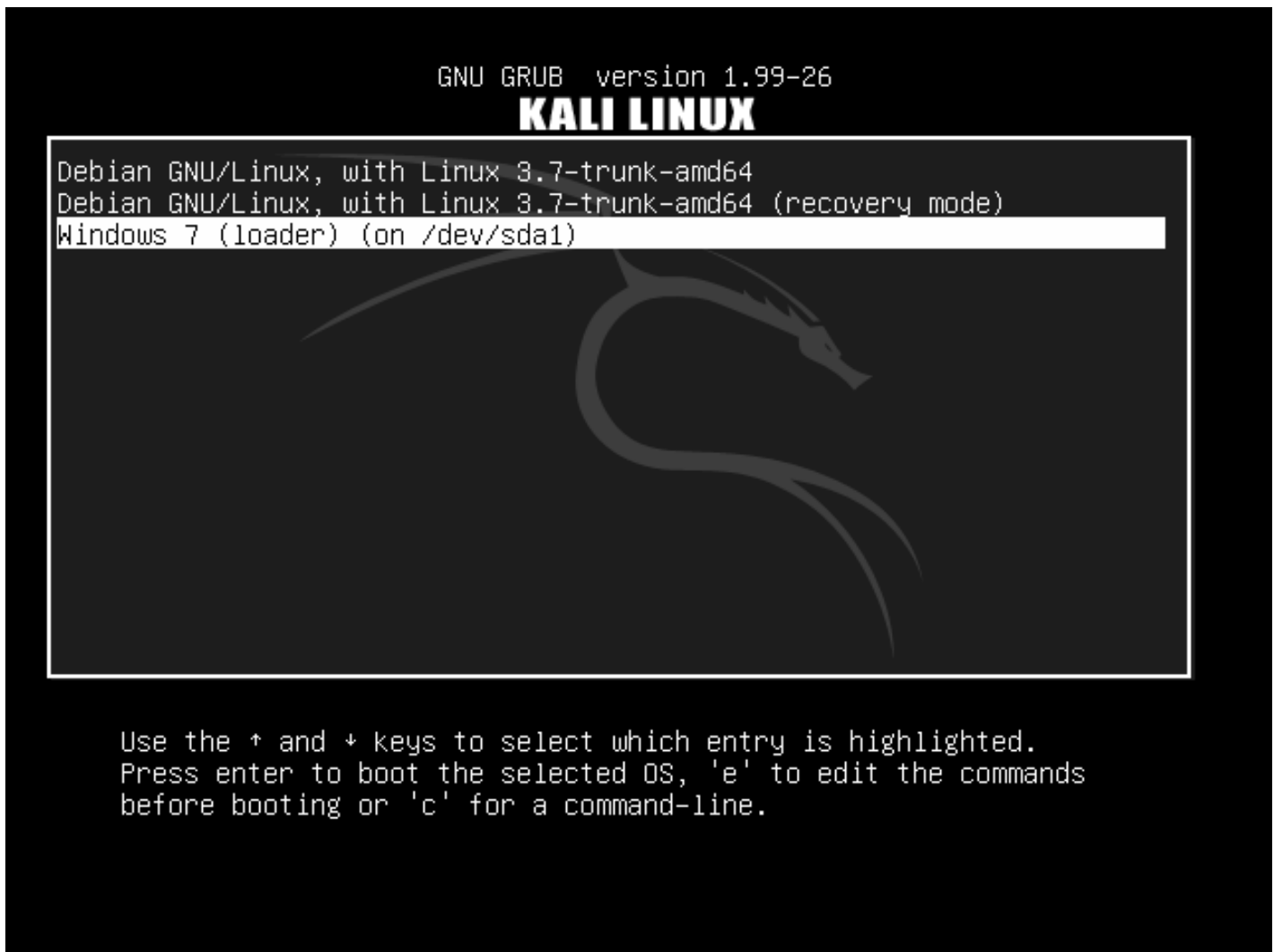


Kali Linux Installation Procedure

1. The installation procedure from this point onwards is similar to a [Kali Linux Hard Disk install](#), until the point of the partitioning, where you need to select “Guided – use the largest continuous free space” that you created earlier with **gparted**.



2. Once the installation is done, reboot. You should be greeted with a GRUB boot menu, which will allow you to boot either into Kali or Windows.



Post Installation

Now that you've completed installing Kali Linux, it's time to customize your system. The [Kali General Use](#) section of our site has more information and you can also find tips on how to get the most out of Kali in our [User Forums](#).

Kali Linux Live USB Install

Booting and installing Kali from a USB stick is our favorite and fastest method of getting up and running. In order to do this, we first need to create the Kali ISO image on a USB drive. If you would like to add persistence to your Kali Linux USB stick, please read the full document before proceeding to create your image.

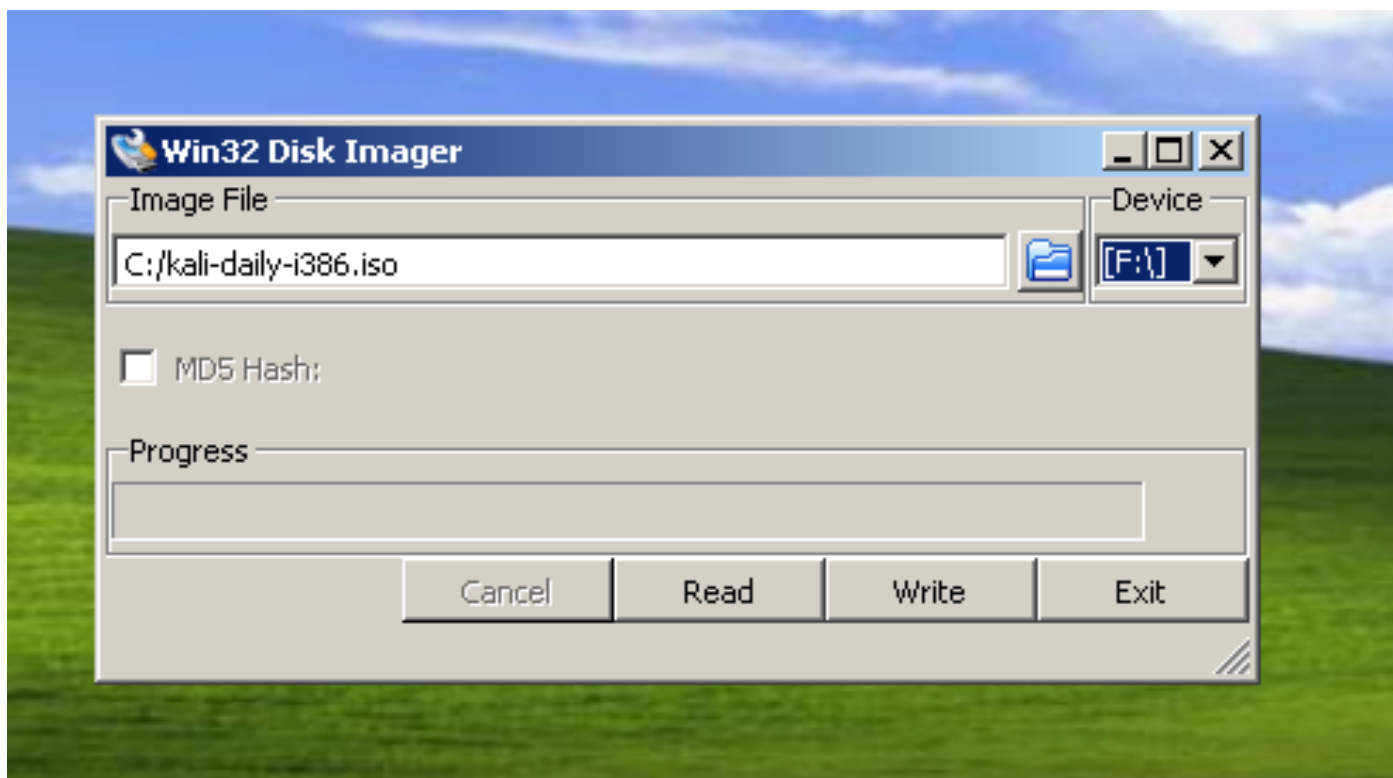
Preparing for the USB copy

1. [Download Kali linux.](#)
2. If running Windows, download [Win32 Disk Imager.](#)
3. No special software is needed for a *nix OS.
4. A USB Key (at least 2GB capacity).

Kali Linux Live USB Install Procedure

Imaging Kali on a Windows Machine

1. Plug your USB stick into your Windows USB port and launch the Win32 Disk Imager software
2. Choose the Kali Linux ISO file to be imaged and verify that the USB drive to be overwritten is the correct one.



3. Once the imaging is complete, safely eject the USB drive from the Windows machine. You can now use the USB device to boot into Kali Linux.

Imaging Kali on a Linux Machine

Creating a bootable Kali Linux USB key in a Linux environment is easy. Once you've downloaded your Kali ISO file, you can use **dd** to copy it over to your USB stick as follows:

WARNING. Although the process of imaging Kali on a USB stick is very easy, you can just as easily destroy arbitrary partitions with **dd** if you do not understand what you are doing. Consider yourself warned.

1. Plug in your USB device to your Linux computer's USB port.
2. Verify the device path of your USB storage with **dmesg**.
3. Proceed to (carefully!) image the Kali ISO file on the USB device:

```
dd if=kali.iso of=/dev/sdb bs=512k
```

That's it, really! You can now boot into a Kali Live / Installer environment using the USB device.

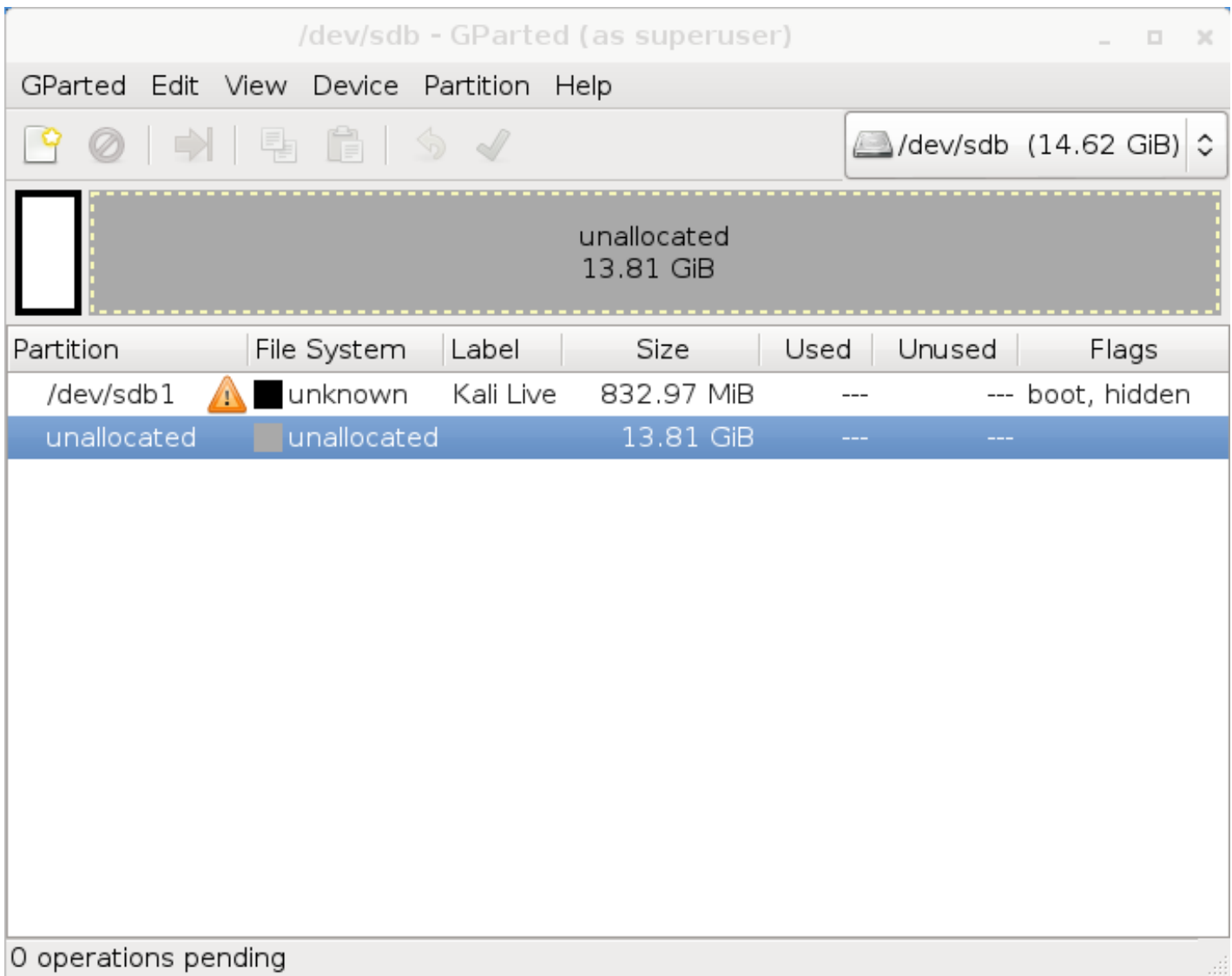
Adding Persistence to Your Kali Live USB

Adding persistence (the ability to save files and changes across live boots) to your Kali Linux image can be very useful in certain situations. To make your Kali Linux USB stick persistent, follow these steps. **In this example, we assume our USB drive is /dev/sdb**. If you want to add persistence, you'll need a larger USB device than we listed in our prerequisites above.

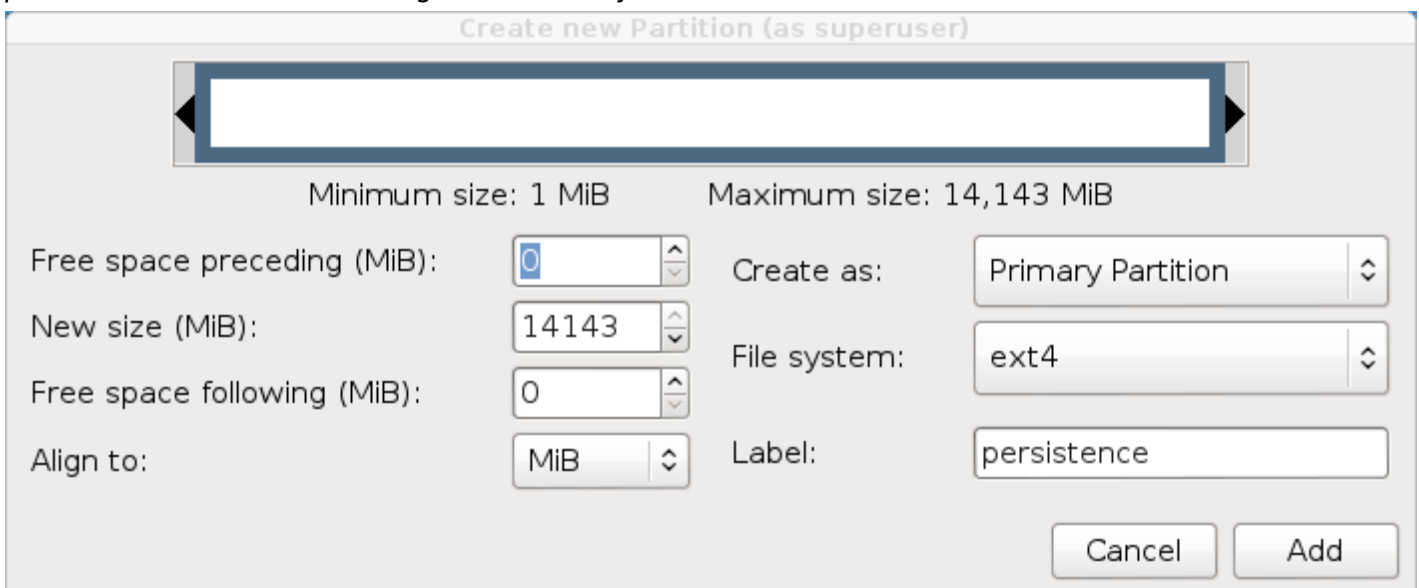
1. Image the Kali Linux ISO to your USB stick as explained above, using the "Linux Method" and **dd**.
2. Create and format an additional partition on the USB stick. In our example, we use **gparted** by invoking:

```
gparted /dev/sdb
```

3. Your current partitioning scheme should look similar to this:



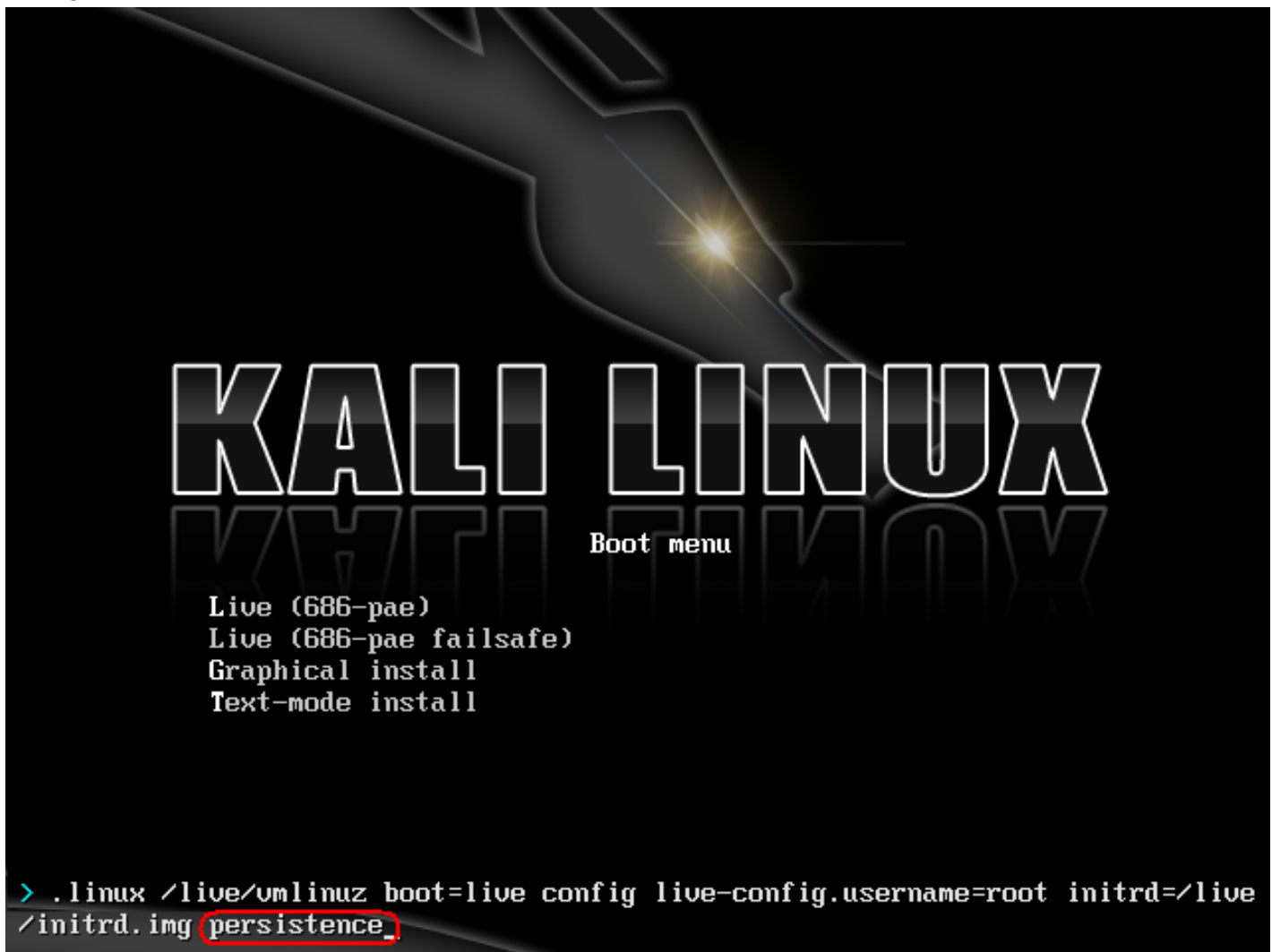
- Proceed to format a new partition of your desired size to be used for persistence. In our example, we used all the remaining space available. Make sure the volume label of the newly created partition is *persistence*, and format it using the *ext4* filesystem.



5. Once the process is complete, mount your persistence USB partition using the following commands:

```
mkdir /mnt/usb  
mount /dev/sdb2 /mnt/usb  
echo "/ union" >> /mnt/usb/persistence.conf  
umount /mnt/usb
```

6. Plug the USB stick into the computer you want to boot up. Make sure your BIOS is set to boot from your USB device. When the Kali Linux boot screen is displayed, select “Live boot” from the menu (don’t press enter), and press the **tab** button. This will allow you to edit the boot parameters. Add the word “persistence” to the end of the boot parameter line each time you want to mount your persistent storage.



Kali Linux Encrypted Disk Install

At times, we have sensitive data we would prefer to encrypt using full disk encryption. With the Kali Installer, you can initiate an LVM encrypted install on either Hard Disk or USB drives. The installation procedure is very similar to a “normal Kali Linux Install”, with the exception of choosing an Encrypted LVM partition during the installation process.

Kali Linux Encrypted Installation Requirements

Installing Kali Linux on your computer is an easy process. First, you’ll need compatible computer hardware. The hardware requirements are minimal as listed below, though better hardware will naturally provide better performance. The i386 images have a default [PAE](#) kernel, so you can run them on systems with over 4GB of RAM. [Download Kali Linux](#) and either burn the ISO to DVD, or [prepare a USB stick with Kali Linux Live](#) as the installation medium.

Installation Prerequisites

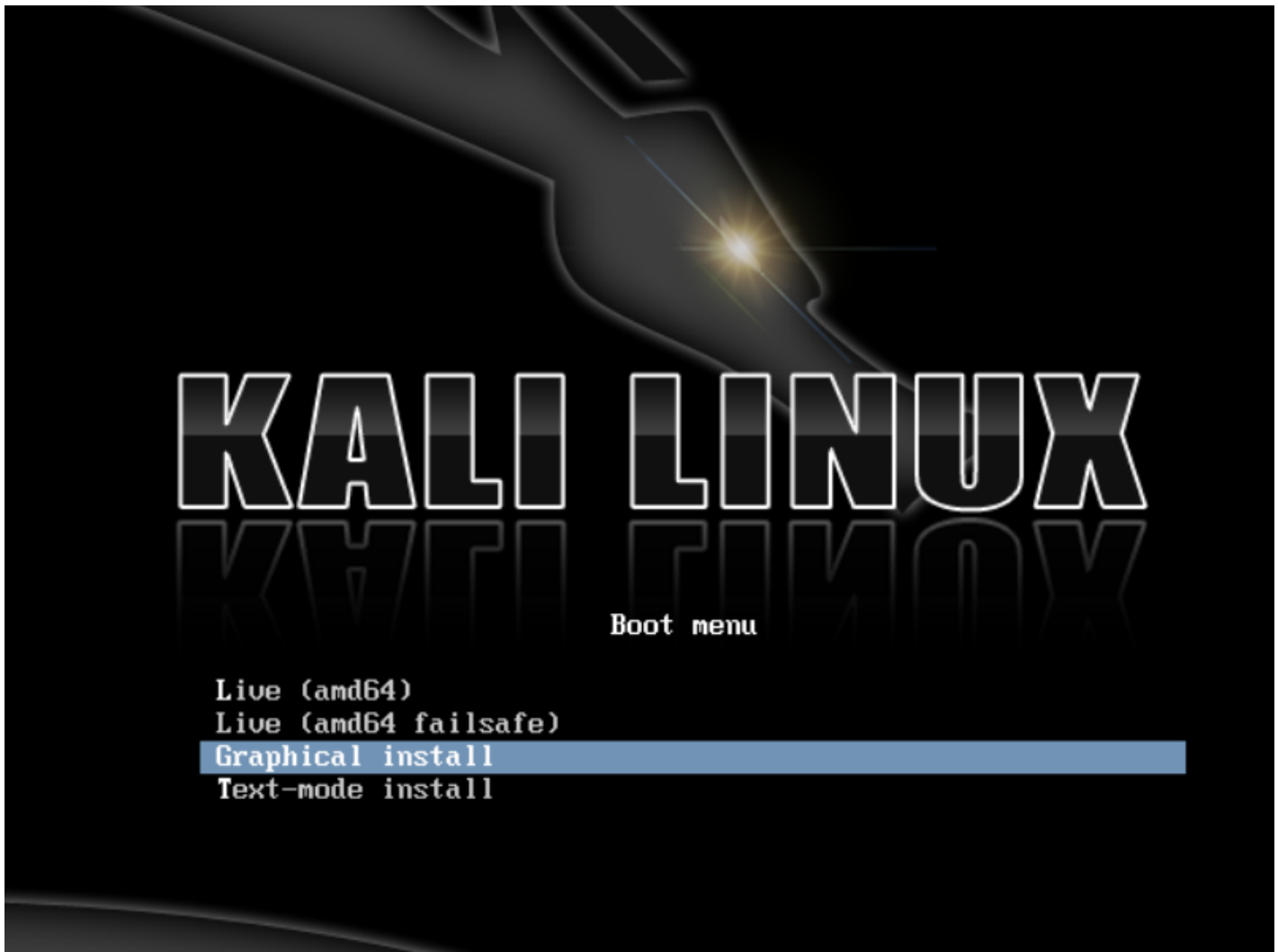
- A minimum of 8 GB disk space for the Kali Linux install.
- For i386 and amd64 architectures, a minimum of 512MB RAM.
- CD-DVD Drive / USB boot support

Preparing for the Installation

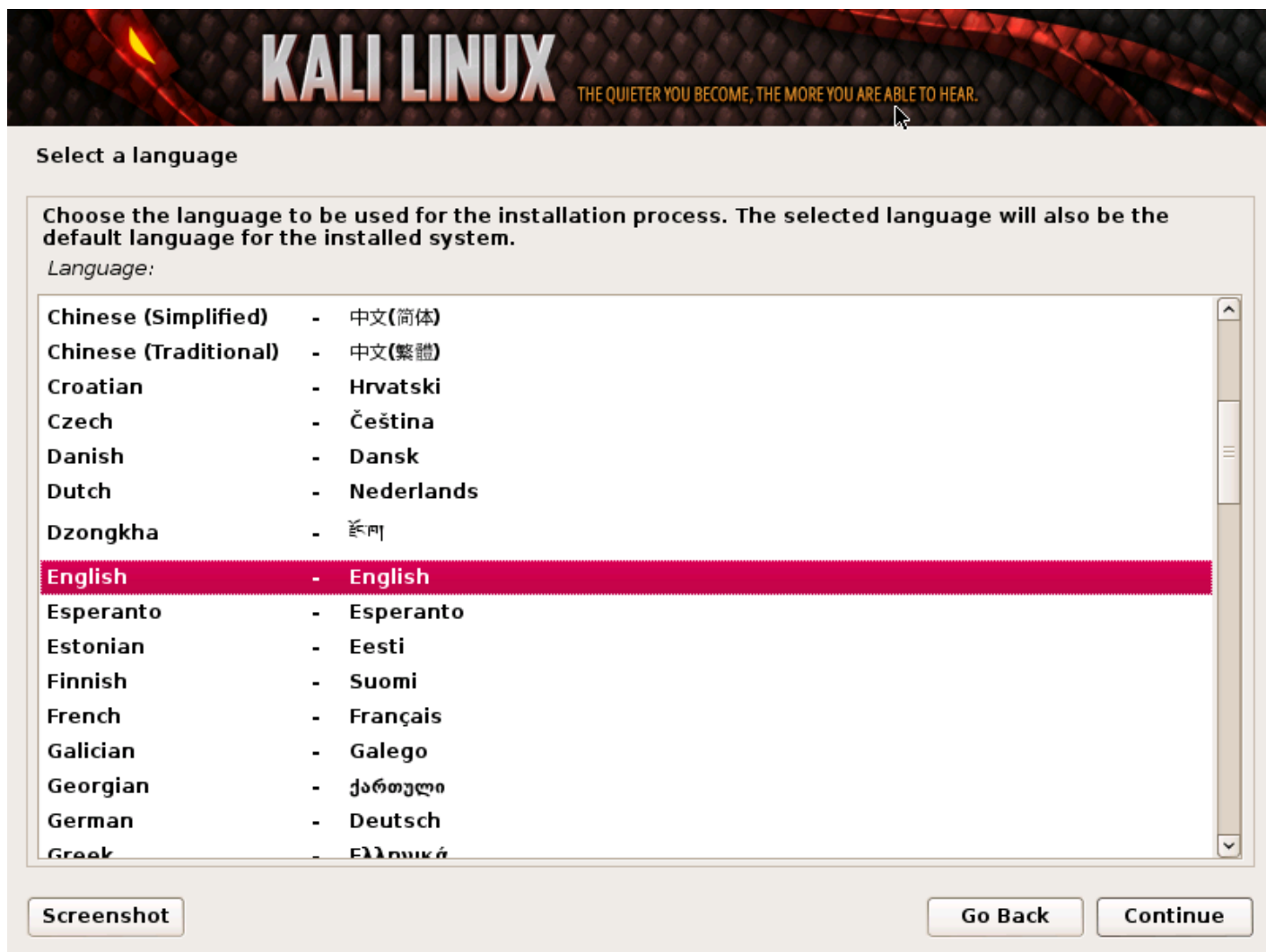
1. [Download Kali linux](#).
2. Burn The Kali linux ISO to DVD or [Image Kali Linux Live to USB](#).
3. Ensure that your computer is set to boot from CD / USB in your BIOS.

Kali Linux Installation Procedure

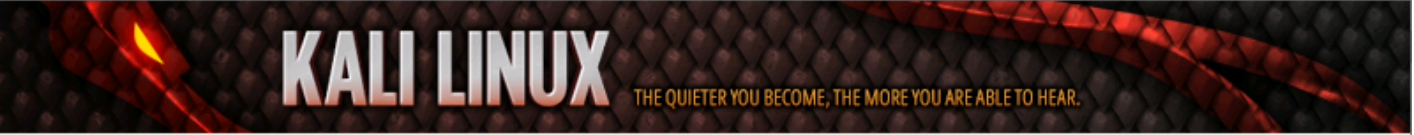
1. To start your installation, boot with your chosen installation medium. You should be greeted with the Kali Linux boot menu. Choose a *Graphical* or a *Text-Mode* install. In this example, we chose a GUI install.



2. Select your preferred language and then your country location. You'll also be prompted to configure your keyboard with the appropriate keymap.



- The installer will copy the image to your hard disk, probe your network interfaces, and then prompt you to enter a hostname for your system. In the example below, we've entered "kali" as the hostname.



Configure the network

Please enter the hostname for this system.

The hostname is a single word that identifies your system to the network. If you don't know what your hostname should be, consult your network administrator. If you are setting up your own home network, you can make something up here.

Hostname:

[Screenshot](#) [Go Back](#) [Continue](#)

4. Enter a robust password for the root account.



Set up users and passwords

You need to set a password for 'root', the system administrative account. A malicious or unqualified user with root access can have disastrous results, so you should take care to choose a root password that is not easy to guess. It should not be a word found in dictionaries, or a word that could be easily associated with you.

A good password will contain a mixture of letters, numbers and punctuation and should be changed at regular intervals.

The root user should not have an empty password. If you leave this empty, the root account will be disabled and the system's initial user account will be given the power to become root using the "sudo" command.

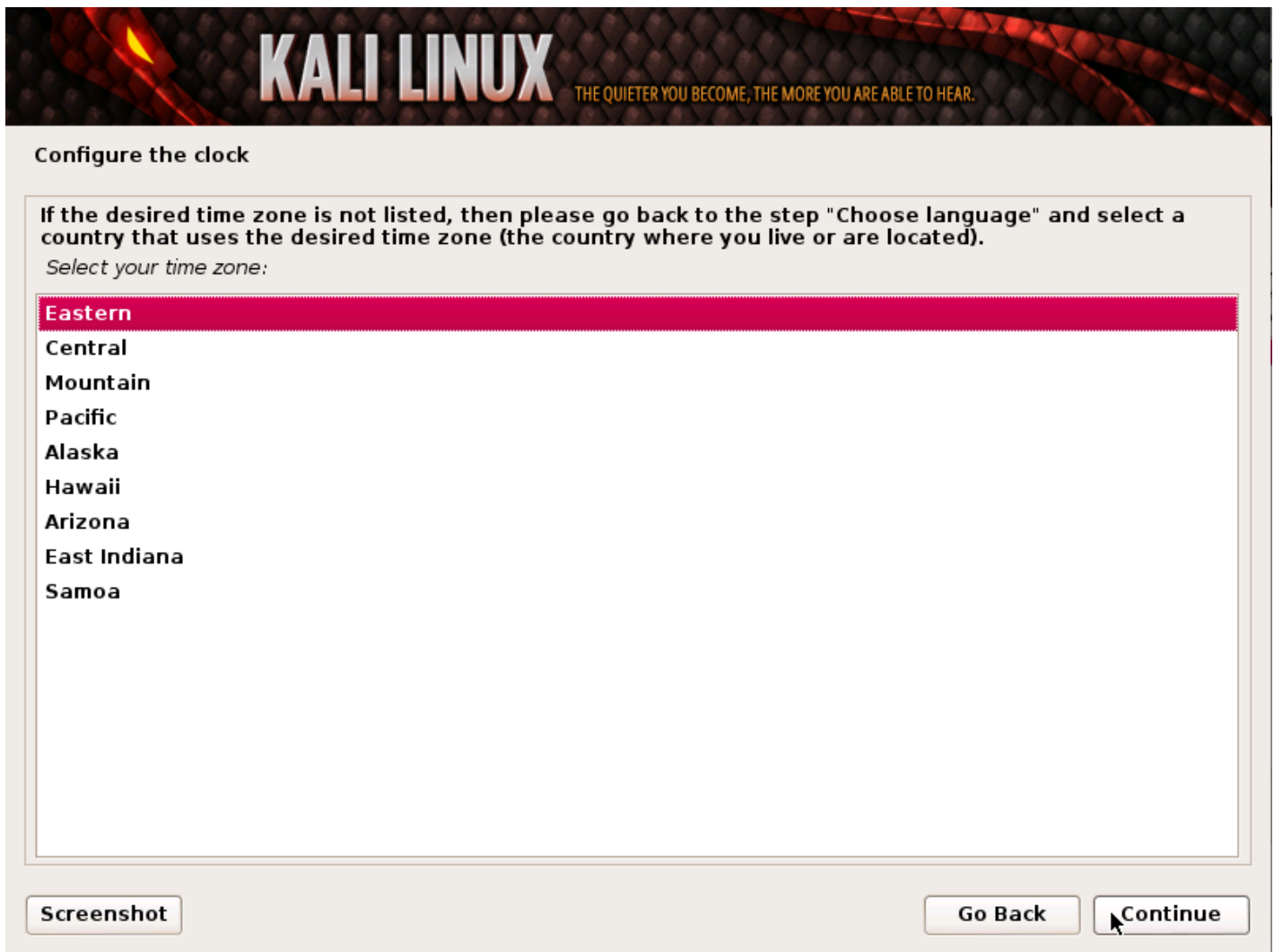
Note that you will not be able to see the password as you type it.

Root password:

Please enter the same root password again to verify that you have typed it correctly.

Re-enter password to verify:

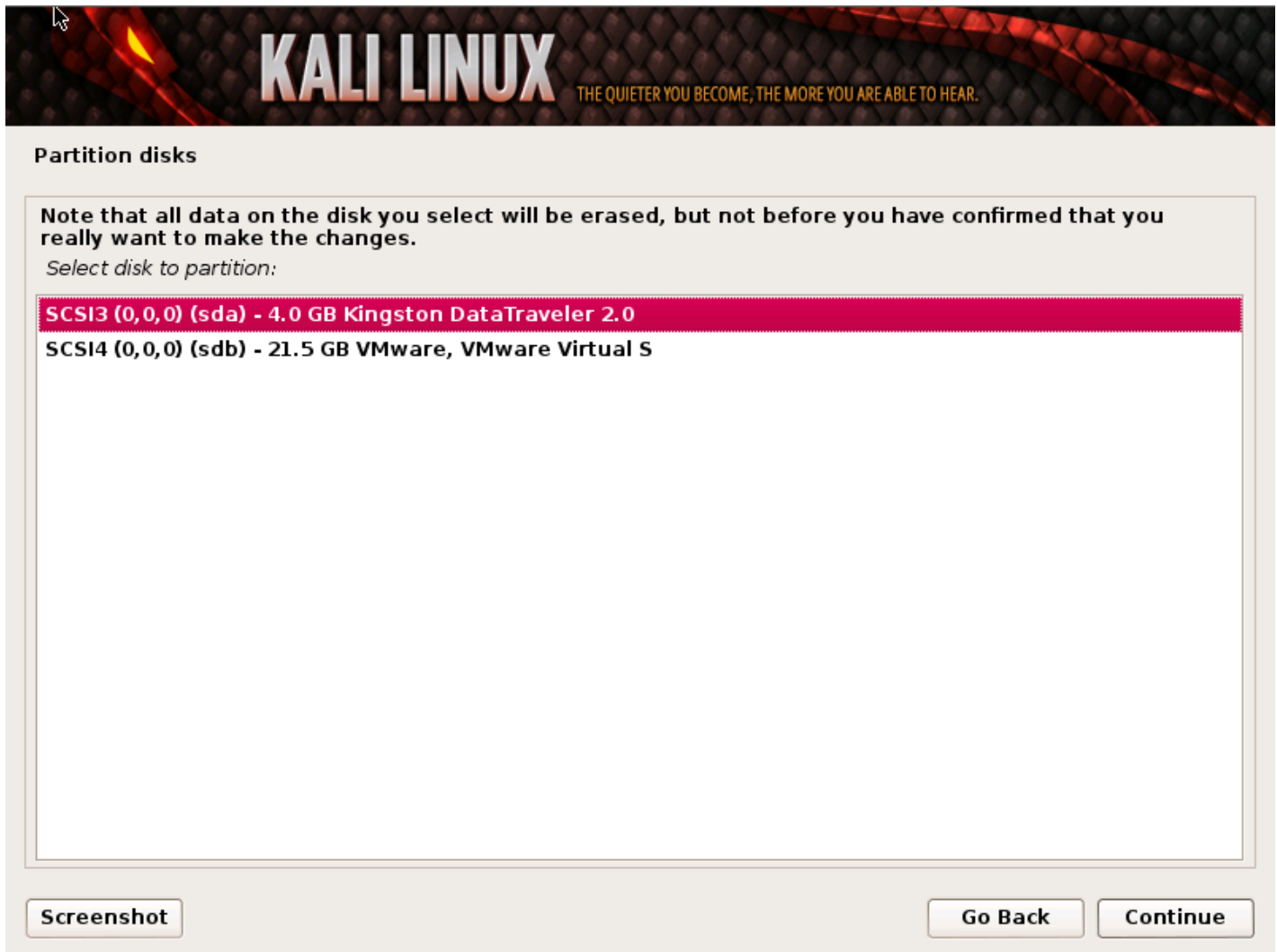
5. Next, set your time zone.



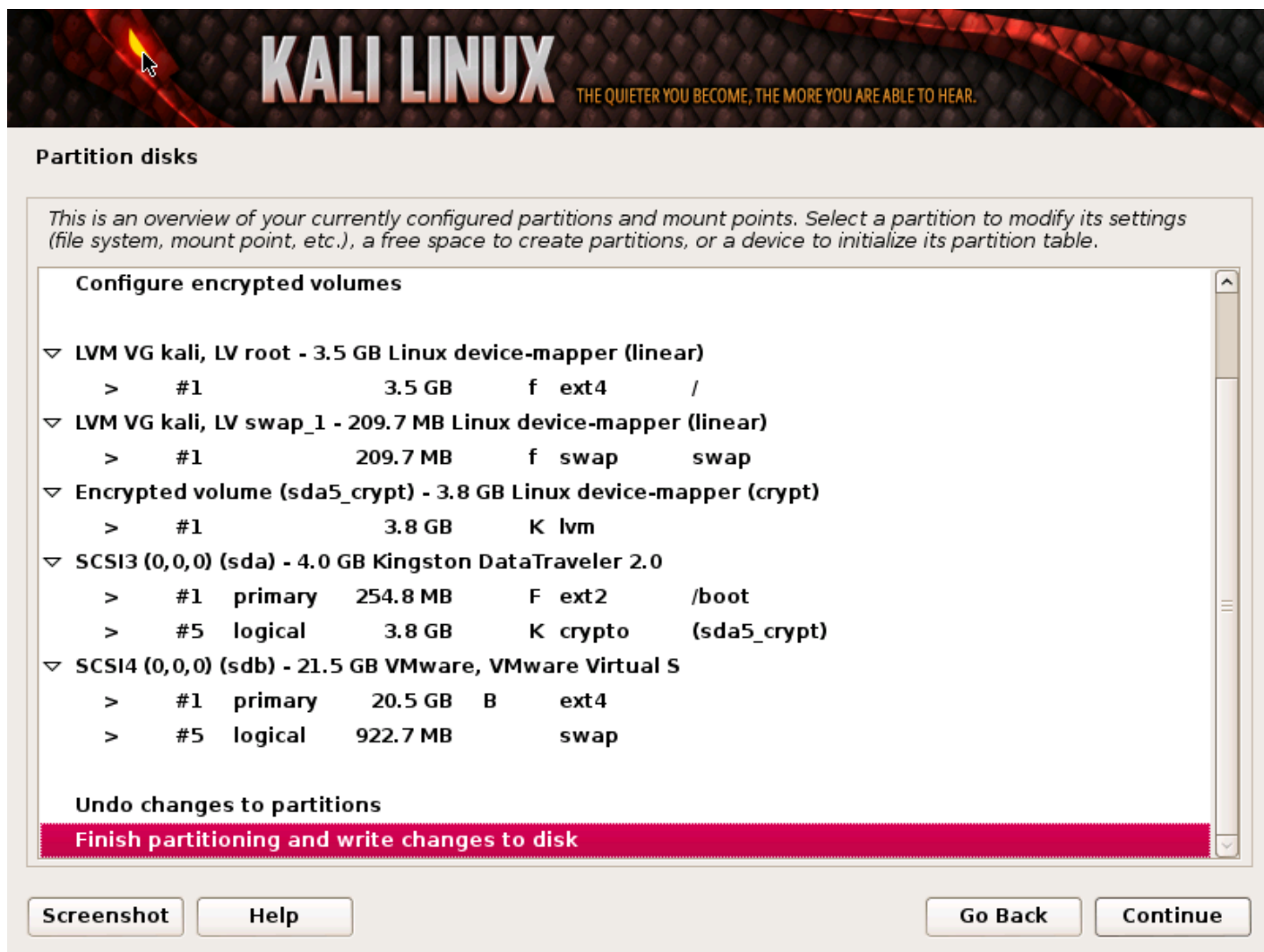
- The installer will now probe your disks and offer you four choices. For an Encrypted LVM install, choose the **"Guided - use entire disk and set up encrypted LVM"** option as shown below.



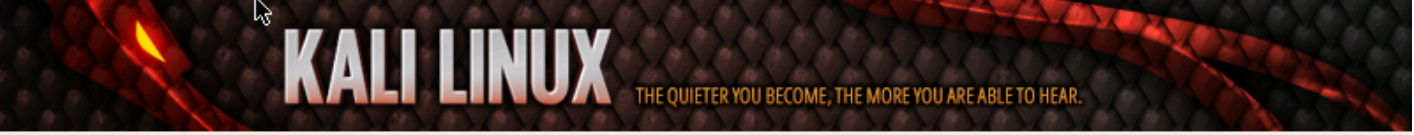
7. Choose the destination drive to install Kali. In this case, we chose a USB drive destination. We will use this USB drive to boot an encrypted instance of Kali.



8. Confirm your partitioning scheme and continue the installation.



9. Next, you will be asked for an encryption password. You will need to remember this password and use it each time to boot the encrypted instance of Kali Linux.



Partition disks

You need to choose a passphrase to encrypt SCSI3 (0,0,0), partition #5 (sda).

The overall strength of the encryption depends strongly on this passphrase, so you should take care to choose a passphrase that is not easy to guess. It should not be a word or sentence found in dictionaries, or a phrase that could be easily associated with you.

A good passphrase will contain a mixture of letters, numbers and punctuation. Passphrases are recommended to have a length of 20 or more characters.

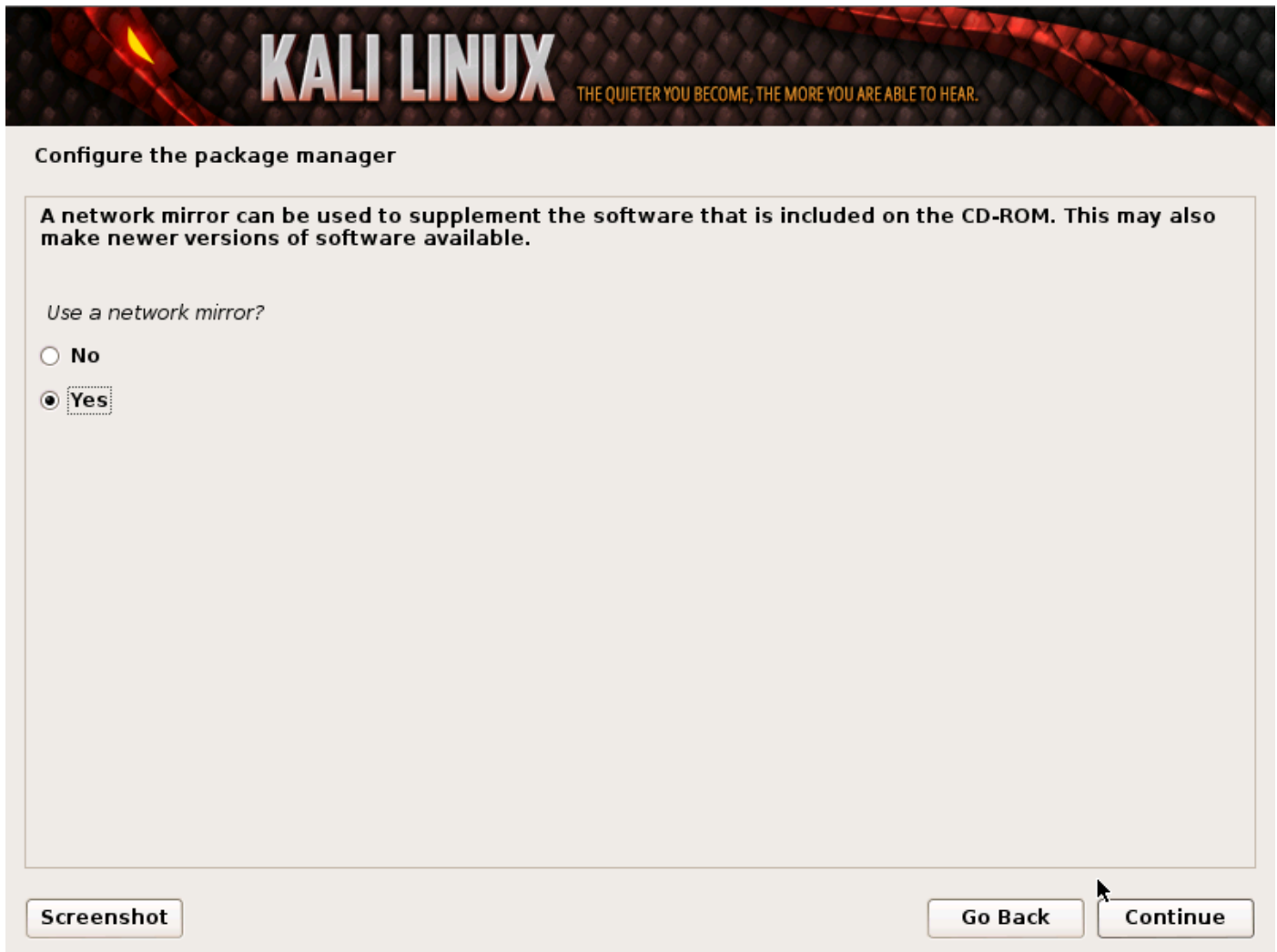
Encryption passphrase:

Please enter the same passphrase again to verify that you have typed it correctly.

Re-enter passphrase to verify:

10. Configure network mirrors. Kali uses a central repository to distribute applications. You'll need to enter any appropriate proxy information as needed.

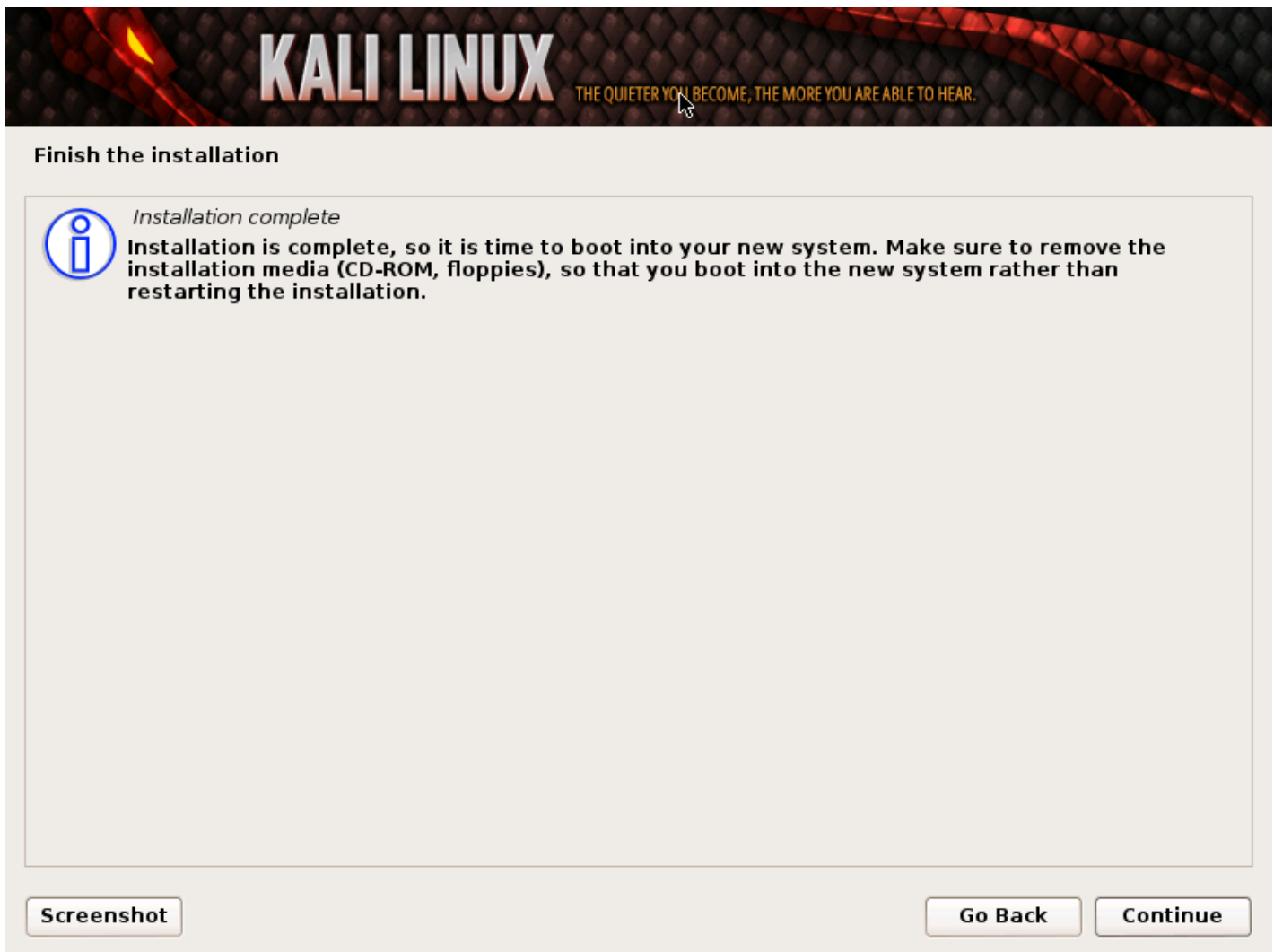
NOTE! If you select "NO" in this screen, you will **NOT** be able to install packages from the Kali repositories.



11. Next, install GRUB.



12. Finally, click *Continue* to reboot into your new Kali installation. If you used a USB device as a destination drive, make sure you enable booting from USB devices in your BIOS. You will be asked for the encryption password you set earlier on every boot.



Post Installation

Now that you've completed installing Kali Linux, it's time to customize your system. The [Kali General Use](#) section of our site has more information and you can also find tips on how to get the most out of Kali in our [User Forums](#).

04. Kali Linux Network Installs

Kali Linux Mini ISO Install

Kali Mini ISO Install

The Kali mini ISO is a convenient way to install a minimal Kali system and install it “from scratch”. The mini install ISO will download all required packages from our repositories, meaning you need to have a fast Internet connection to use this installation method.

Installation Prerequisites

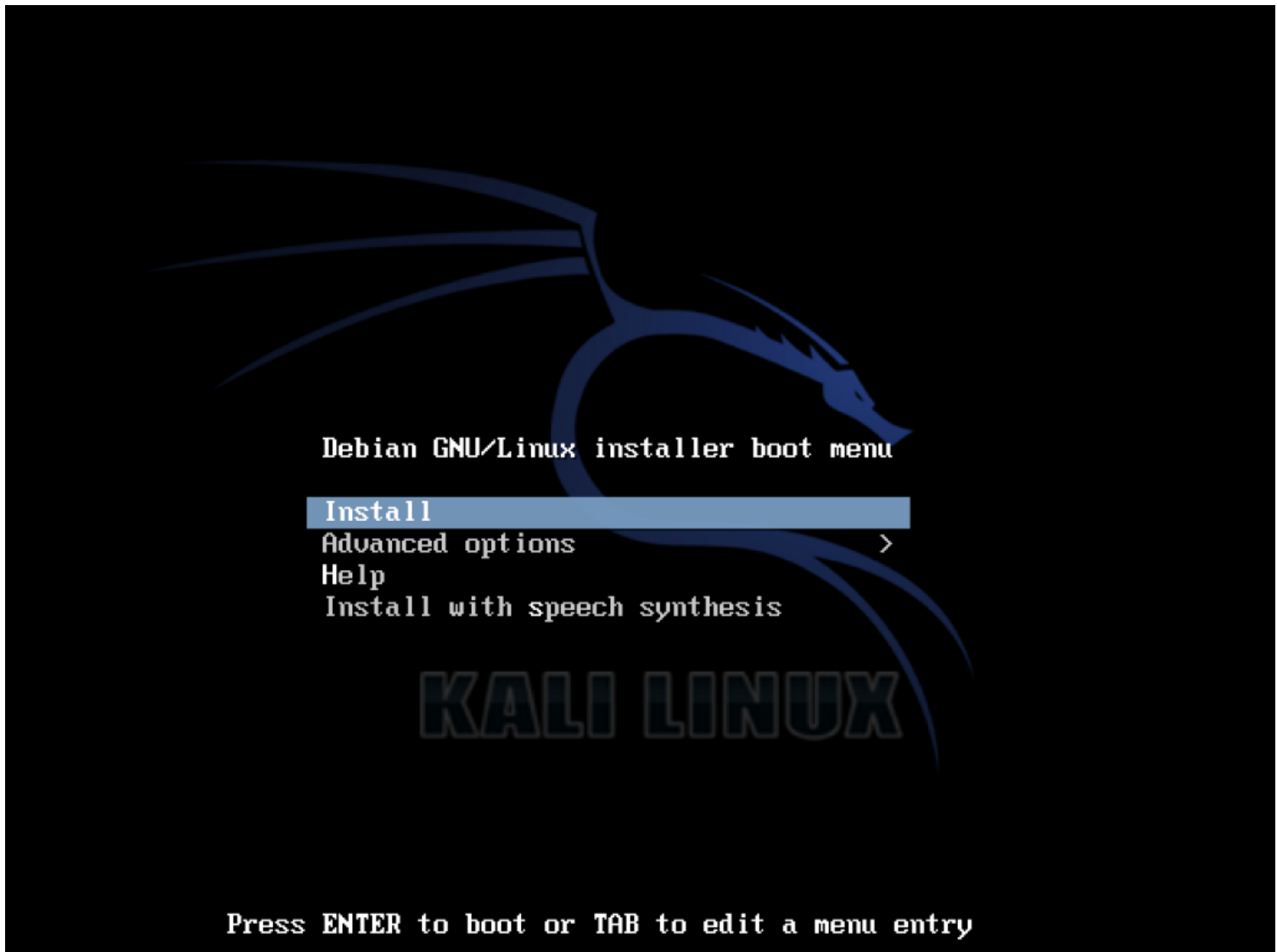
- A minimum of 8 GB disk space for the Kali Linux install.
- For i386 and amd64 architectures, a minimum of 512MB RAM.
- CD-DVD Drive / USB boot support

Preparing for the Installation

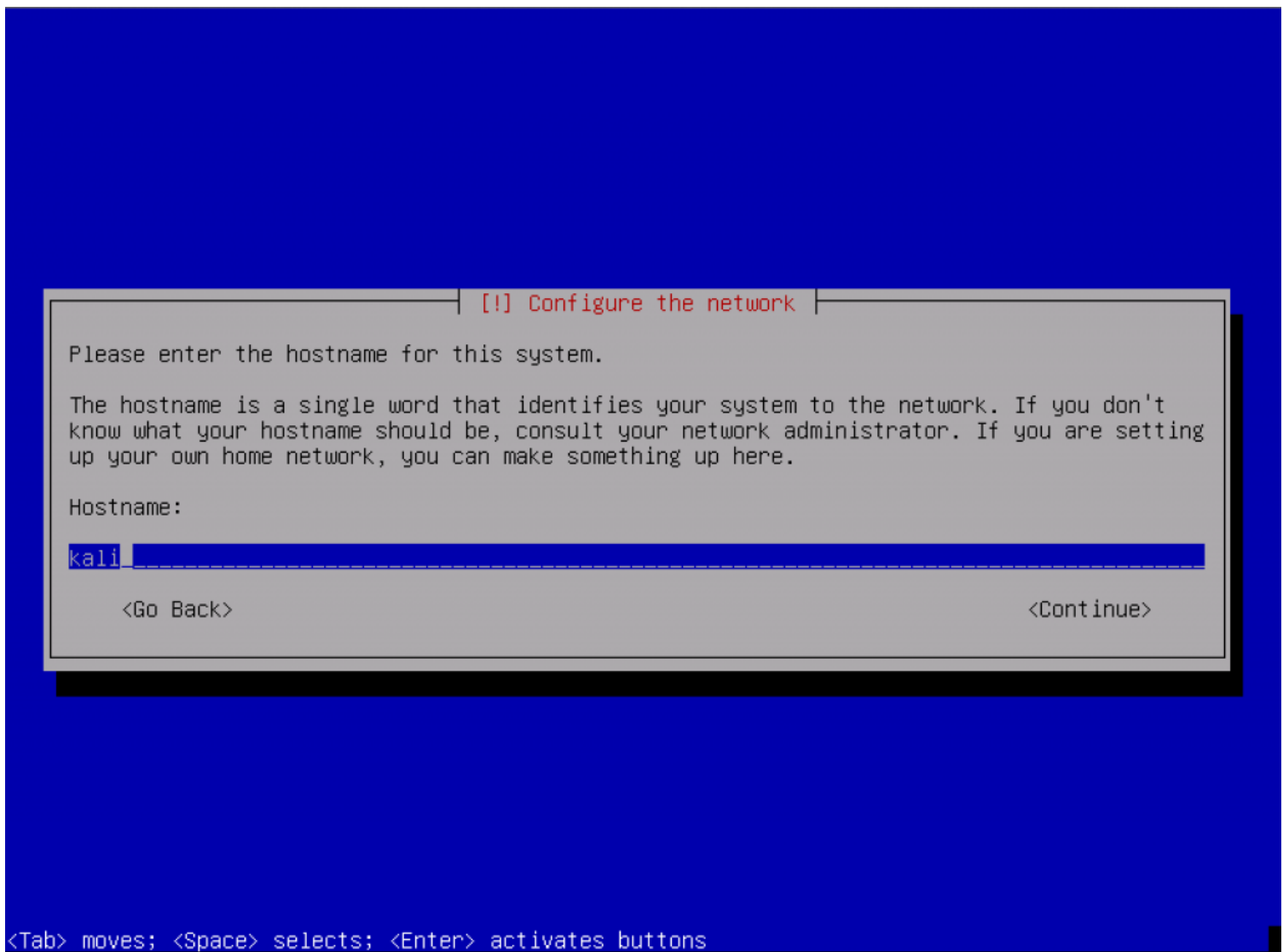
1. [Download the Kali mini ISO.](#)
2. Burn The Kali Linux ISO to DVD or [Image Kali Linux Live to USB.](#)
3. Ensure that your computer is set to boot from CD / USB in your BIOS.

Kali Linux Installation Procedure

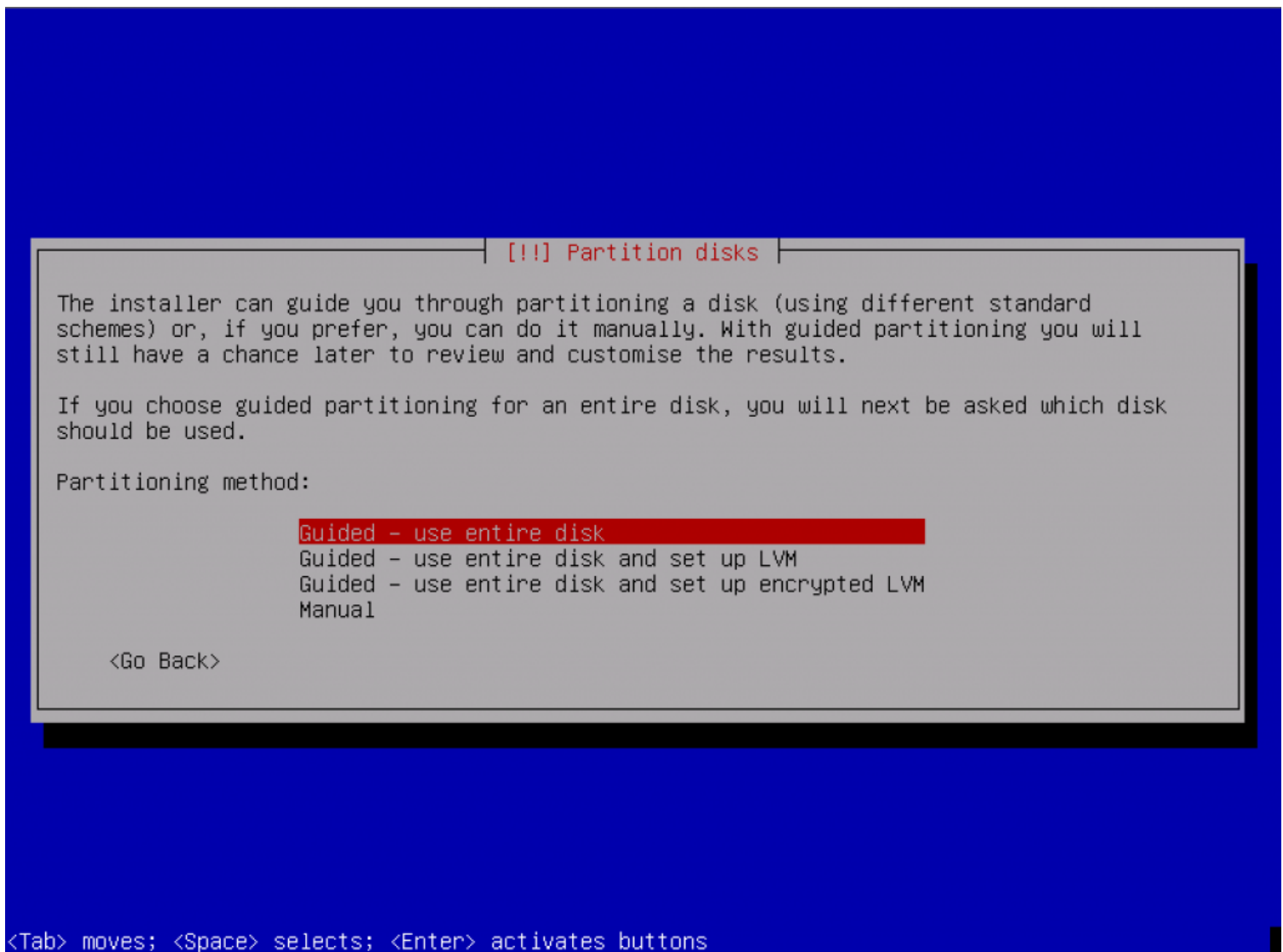
When you first boot the mini ISO, you will be presented with a small boot menu with various options. For this article, we will simply be doing a basic install.



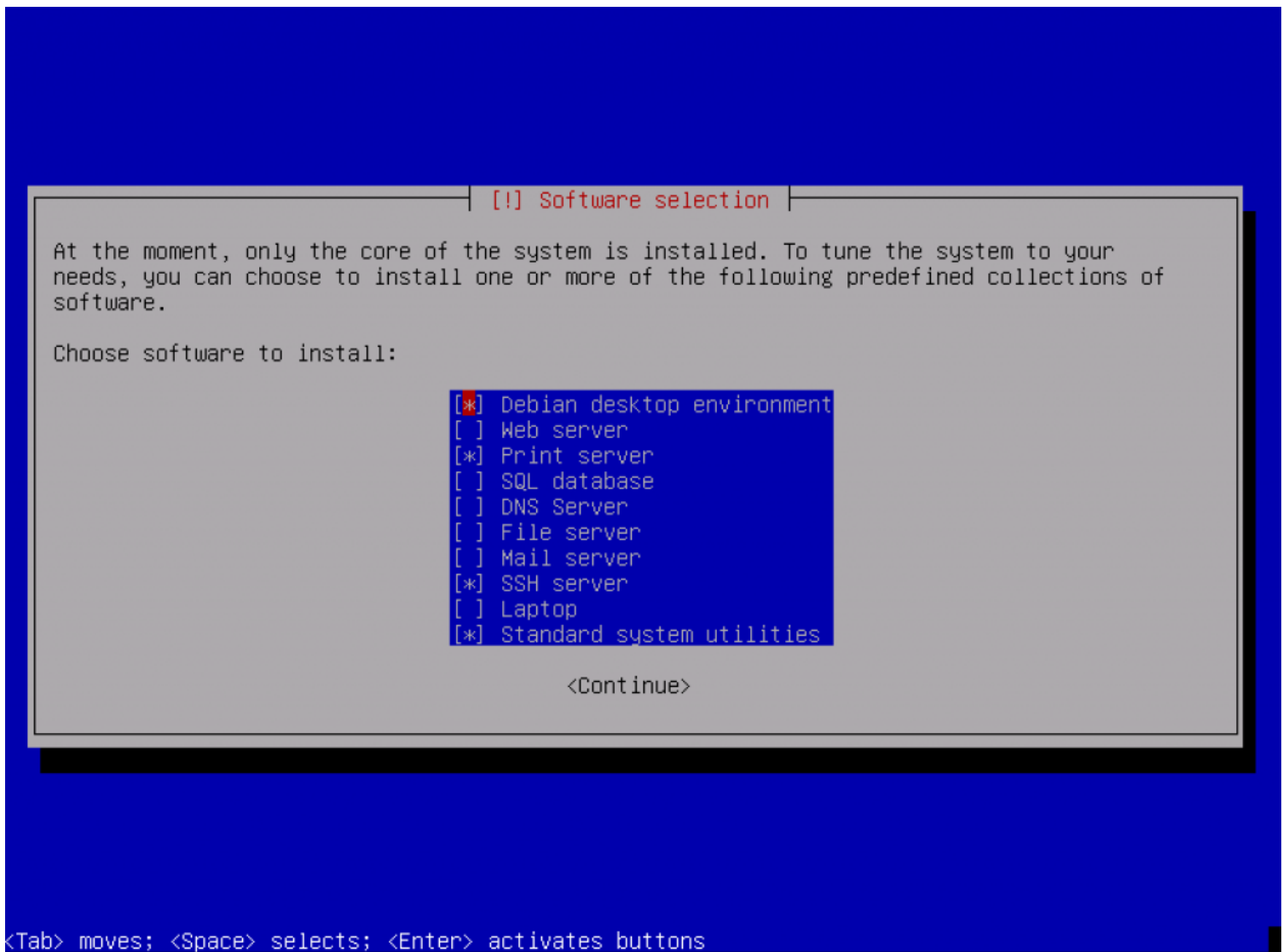
You will next be prompted for various things such as your language and keyboard type, then you will need to select a hostname for your installation. We will stick with the default of *kali*.



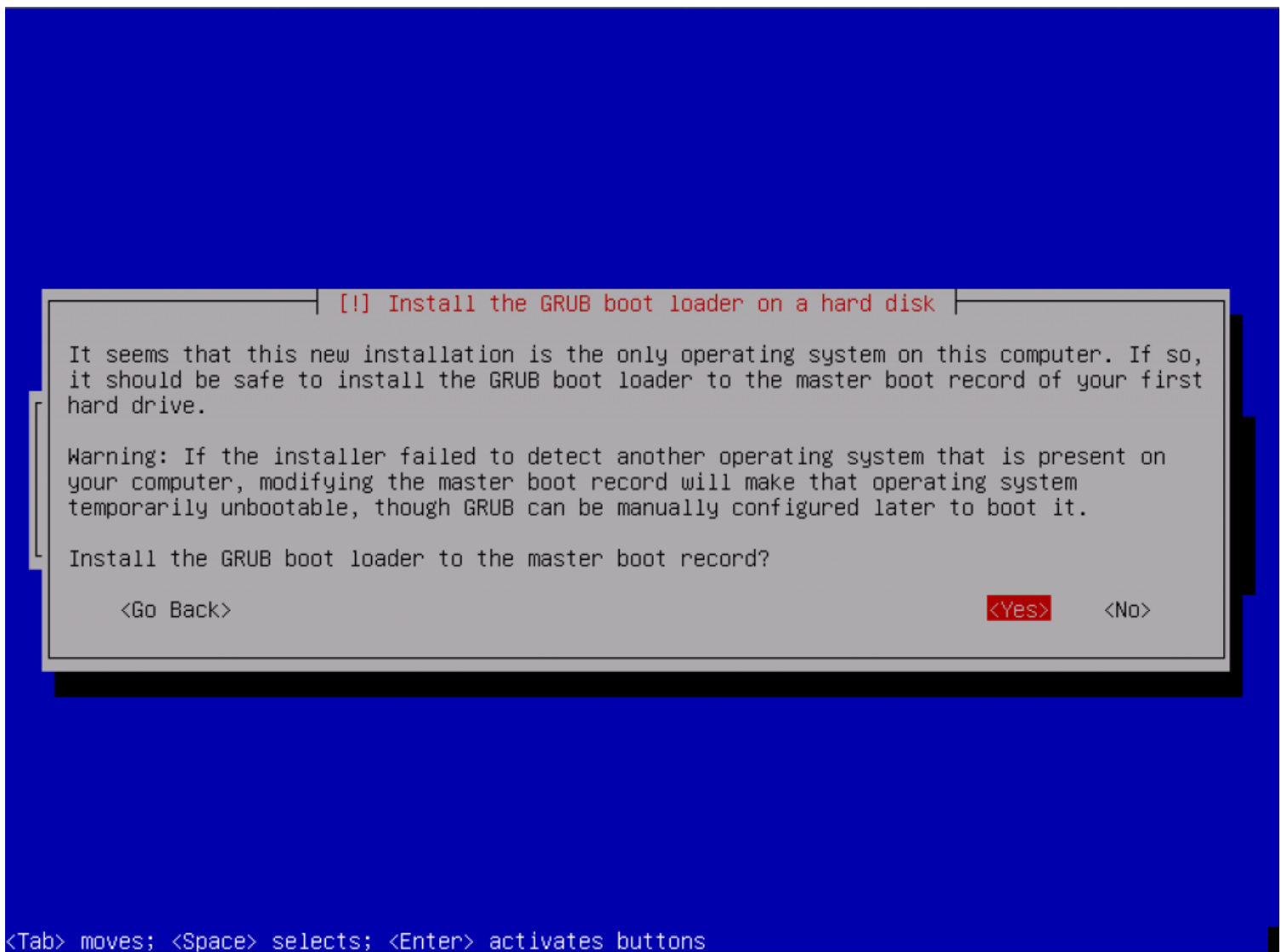
Next, you will need to select your time zone, then you'll be shown the partition options. To get up and running quickly, we will use 'Guided - use entire disk' and follow the prompts all the way through to create the new partitioning setup.



In order to reduce network bandwidth, a small subset of packages will be selected by default. If you wish to add different services or features, this is the area you would make your selections.



At this point, the installer will download all of the packages it requires and install them on the system. Depending on your Internet connectivity speed, this could take some time. Eventually, you will finally be prompted to install GRUB to finish the installation.



Post Installation

Now that you've completed installing Kali Linux, it's time to customize your system. The [Kali General Use](#) section of our site has more information and you can also find tips on how to get the most out of Kali in our [User Forums](#).

Kali Linux Network PXE Install

Setup a PXE Server

Booting and installing Kali over the network ([PXE](#)) can be useful from a single laptop install with no CDROM or USB ports, to enterprise deployments supporting pre-seeding of the Kali installation.

First, we need to install *dnsmasq* to provide the DHCP/TFTP server and then edit the *dnsmasq.conf* file.

```
apt-get install dnsmasq
nano /etc/dnsmasq.conf
```

In *dnsmasq.conf*, enable DHCP, TFTP and PXE booting and set the *dhcp-range* to match your environment. If needed you can also define your gateway and DNS servers with the *dhcp-option* directive as shown below:

```
interface=eth0
dhcp-range=192.168.101.100,192.168.101.200,12h
dhcp-boot=pxelinux.0
enable-tftp
tftp-root=/tftpboot/
dhcp-option=3,192.168.101.1
dhcp-option=6,8.8.8.8,8.8.4.4
```

With the edits in place, the *dnsmasq* service needs to be restarted in order for the changes to take effect.

```
service dnsmasq restart
```

Download Kali PXE Netboot Images

Now, we need to create a directory to hold the Kali Netboot image and download the image we wish to serve from the Kali repos.

```
mkdir -p /tftpboot
cd /tftpboot
# for 64 bit systems:
wget http://repo.kali.org/kali/dists/kali/main/installer-amd64/current/images/netboot/netboot.tar.gz
# for 32 bit systems:
wget http://repo.kali.org/kali/dists/kali/main/installer-i386/current/images/netboot/netboot.tar.gz
tar xzpf netboot.tar.gz
rm netboot.tar.gz
```

Configure Target to Boot From Network

With everything configured, you can now boot your target system and configure it to boot from the network. It should get an IP address from your PXE server and begin booting Kali.

05. Kali Linux General Use

Kali Linux sources.list repositories

We've seen many people break their Kali Linux installations by following unofficial advice, or arbitrarily populating their sources.list file with unneeded repositories. The following post aims to clarify what repositories should exist in sources.list, and when they should be used.

Any additional repositories added to the Kali sources.list file will most likely BREAK YOUR KALI LINUX INSTALL.

Regular repositories

On a standard, clean install of Kali Linux, you should have the following two entries present:

```
deb http://http.kali.org/kali kali main non-free contrib
deb http://security.kali.org/kali-security kali/updates main contrib non-free
```

Source repositories

In case you require source packages, you might also want to add the following repositories as well:

```
deb-src http://http.kali.org/kali kali main non-free contrib
deb-src http://security.kali.org/kali-security kali/updates main contrib non-free
```

Bleeding Edge repositories

If you have a need for bleeding edge repositories, you can add the following entry. Do not add this repo "for the heck of it" - it's called "bleeding edge" for a reason. Packages in this repository are NOT manually maintained (they are auto-generated), and are low priority in general.

```
deb http://repo.kali.org/kali kali-bleeding-edge main
#deb-src http://repo.kali.org/kali kali-bleeding-edge main
```

Install NVIDIA Drivers on Kali

This document explains how to make use of NVIDIA video hardware and install the drivers on a Kali Linux system. The first step is to fully update your Kali Linux system and make sure you have the kernel headers installed.

```
apt-get update
apt-get install -y linux-headers-$(uname -r)
```

Next, download the latest NVIDIA driver for your architecture and video card [here](#). To locate your NVIDIA card model, execute the following command.

```
root@kali:~# lspci | grep -i vga
02:00.0 VGA compatible controller: NVIDIA Corporation GT218 [GeForce G210M] (rev a2)
03:00.0 VGA compatible controller: NVIDIA Corporation C79 [GeForce 9400M G] (rev b1)
```

The next step is to disable the *nouveau* driver. [Nouveau](#) is an open source NVIDIA driver project, however it lacks the 3D graphics acceleration needed to run Cuda pentest tools.

```
sed 's/quiet/quiet nouveau.modeset=0/g' -i /etc/default/grub
update-grub
reboot
```

Once the system has rebooted and you are looking at the GDM login screen, press CTRL+ALT+F1 in order to get

to a TTY, which will be a black screen with a login prompt. We need to login as root and stop the gdm3 service as follows.

```
service gdm3 stop
```

If you are on a 64-bit Kali system, you may want to install the *ia32-libs* package in order to allow the NVIDIA installer to install the 32-bit libraries, although this is optional. If you choose not to do it, simply select **no** when the installer asks if you want to install the 32-bit libraries.

```
dpkg --add-architecture i386  
apt-get update  
apt-get install ia32-libs
```

Assuming you downloaded the NVIDIA driver to your */root/* directory, we need to give it executable permissions and run it.

```
chmod 755 NVIDIA-Linux-x86_64-310.44.run  
./NVIDIA-Linux-x86_64-310.44.run
```

Once the installer finishes, you should reboot your machine. Once the system boots back up, you may see a NVIDIA splash screen, which will indicate that the drivers installed correctly, however, this is not always the case. In order to check if the drivers are working properly, execute the following command.

```
root@kali:~# glxinfo | grep -i "direct rendering"  
direct rendering: Yes
```

Although utilizing the Cuda tools included in Kali is beyond the scope of this article, checking to make sure that they are working properly is always a good idea. The following command uses Oclhashcat-plus with some of the example files included in the package.

```
cd /usr/share/oclhashcat-plus/  
./cudaHashcat-plus.bin -t 32 -a 7 example0.hash ?a?a?a?a example.dict  
cudaHashcat-plus v0.14 by atom starting...
```

```
Hashes: 6494 total, 1 unique salts, 6494 unique digests  
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes  
Workload: 256 loops, 80 accel  
Watchdog: Temperature abort trigger set to 90c  
Watchdog: Temperature retain trigger set to 80c  
Device #1: GeForce G210M, 511MB, 1468Mhz, 2MCU  
Device #2: GeForce 9400M G, 253MB, 1100Mhz, 2MCU  
Device #1: Kernel ./kernels/4318/m0000_a1.sm_12.64.ptx  
Device #2: Kernel ./kernels/4318/m0000_a1.sm_11.64.ptx
```

```
Generated dictionary stats for example.dict: 1210228 bytes, 129988 words, 129988 keyspace
```

```
27b797965af03466041487f2a455fe52:mo0000  
a48dd0f09abaf64324be83ce86414b5f:ap2300000  
7becb9424f38abff581f6f2a82ff436a:sail00  
1459ccf0940e63051d5a875a88acfaaf:pigi00  
3baa3048651a65d1260eb521ab7c3bc0:ek110  
7a7a8220266f71f54f85685969ce999f:davi0123456789  
98c627ca129e64dfff3bf08fbaab6c86:fire01man
```


As you can see in the output above, the cards are recognized and the passwords are being recovered successfully.

Kali Linux Virtual Box Guest

Should you decide to install Kali Linux within VirtualBox, you will need to follow the instructions below in order to successfully install the Linux Guest Addition tools.

You must use version **4.2.xx or higher** of VirtualBox in order to take advantage of the improvements, including compatibility updates, and enhanced stability of both the core application and the Guest Additions.

Installing VirtualBox Guest Additions in Kali Linux

In order to have proper mouse and screen integration as well as folder sharing with your host system, you will need to install the VirtualBox Guest additions.

Once you have booted into your Kali Linux virtual machine, open a terminal window and issue the following command to install the Linux Kernel headers.

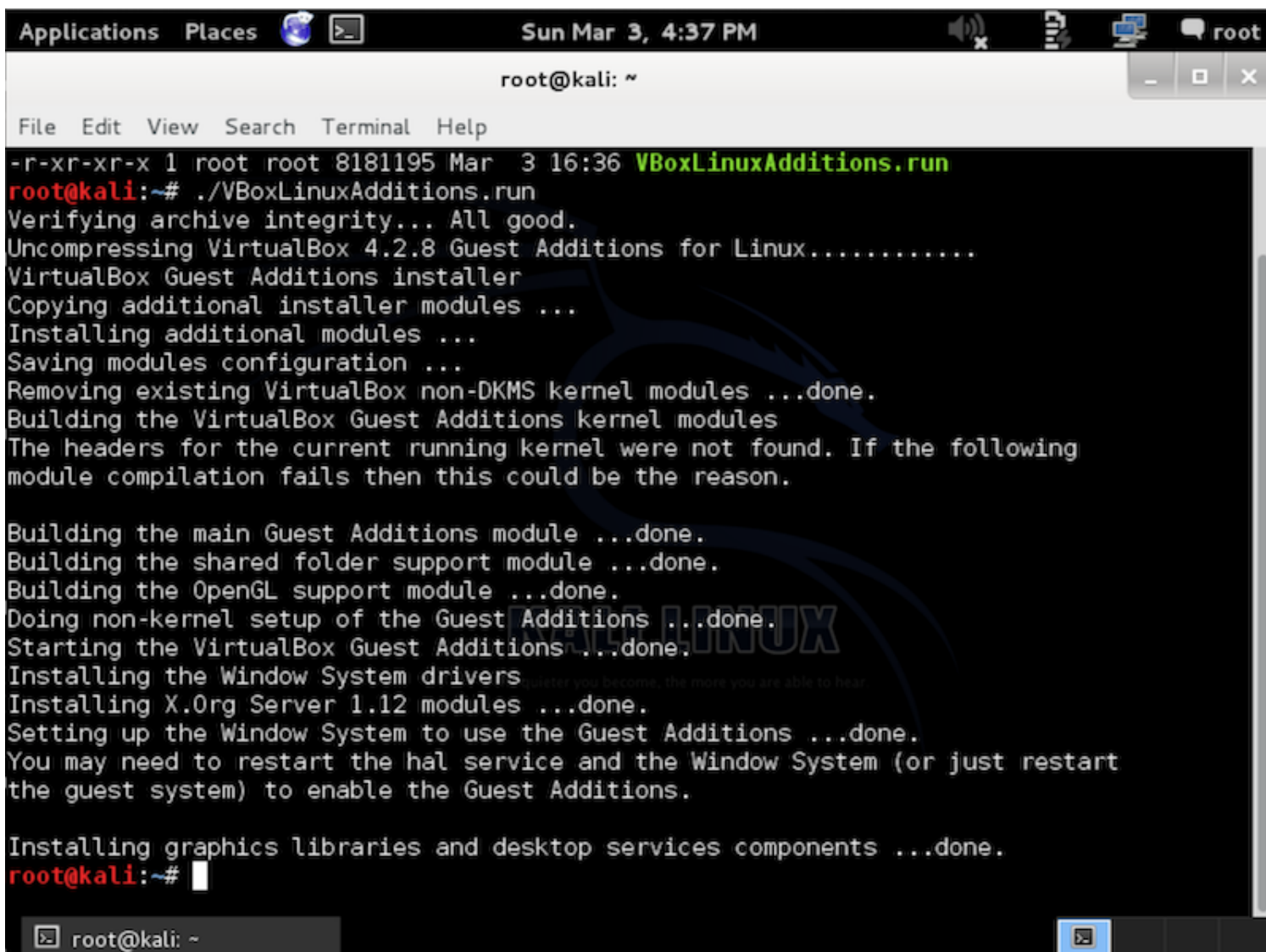
```
apt-get update && apt-get install -y linux-headers-$(uname -r)
```

Once this is complete you can now attach the Guest Additions CD-Rom. This can be done by selecting 'Devices' from the VirtualBox Menu and selecting 'Install Guest Additions.' This will mount the GuestAdditions iso to the virtual CD Drive in your Kali Linux virtual machine. When prompted to autorun the CD, click the Cancel button.



From a terminal window, copy the `VboxLinuxAdditions.run` file from the Guest Additions CD-Rom to a path on your local system ensure it is executable and run the file to begin installation.

```
cp /media/cd-rom/VBoxLinuxAdditions.run /root/  
chmod 755 /root/VBoxLinuxAdditions.run  
cd /root  
./VBoxLinuxAdditions.run
```



```
Applications Places Sun Mar 3, 4:37 PM root
root@kali: ~
File Edit View Search Terminal Help
-r-xr-xr-x 1 root root 8181195 Mar  3 16:36 VBoxLinuxAdditions.run
root@kali:~# ./VBoxLinuxAdditions.run
Verifying archive integrity... All good.
Uncompressing VirtualBox 4.2.8 Guest Additions for Linux.....
VirtualBox Guest Additions installer
Copying additional installer modules ...
Installing additional modules ...
Saving modules configuration ...
Removing existing VirtualBox non-DKMS kernel modules ...done.
Building the VirtualBox Guest Additions kernel modules
The headers for the current running kernel were not found. If the following
module compilation fails then this could be the reason.

Building the main Guest Additions module ...done.
Building the shared folder support module ...done.
Building the OpenGL support module ...done.
Doing non-kernel setup of the Guest Additions ...done.
Starting the VirtualBox Guest Additions...done.
Installing the Window System drivers
Installing X.Org Server 1.12 modules ...done.
Setting up the Window System to use the Guest Additions ...done.
You may need to restart the hal service and the Window System (or just restart
the guest system) to enable the Guest Additions.

Installing graphics libraries and desktop services components ...done.
root@kali:~#
```

Reboot the Kali Linux VM to complete the Guest Additions installation. You should now have full mouse and screen integration as well as the ability to share folders with the host system.

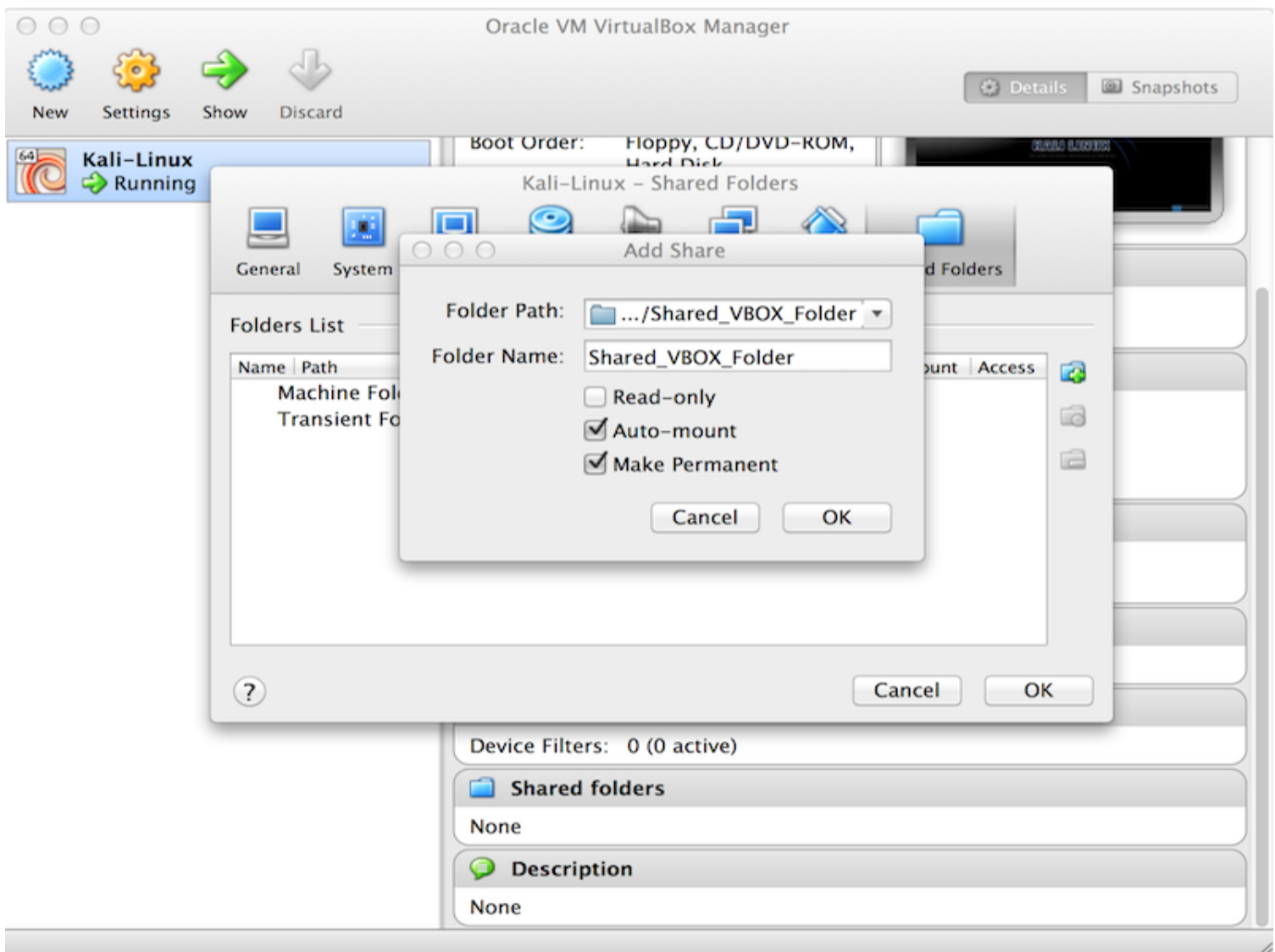
Creating Shared Folders with the Host System

In order to share folders on your host system with your Kali Linux VM, there are a few short steps that need to be completed.

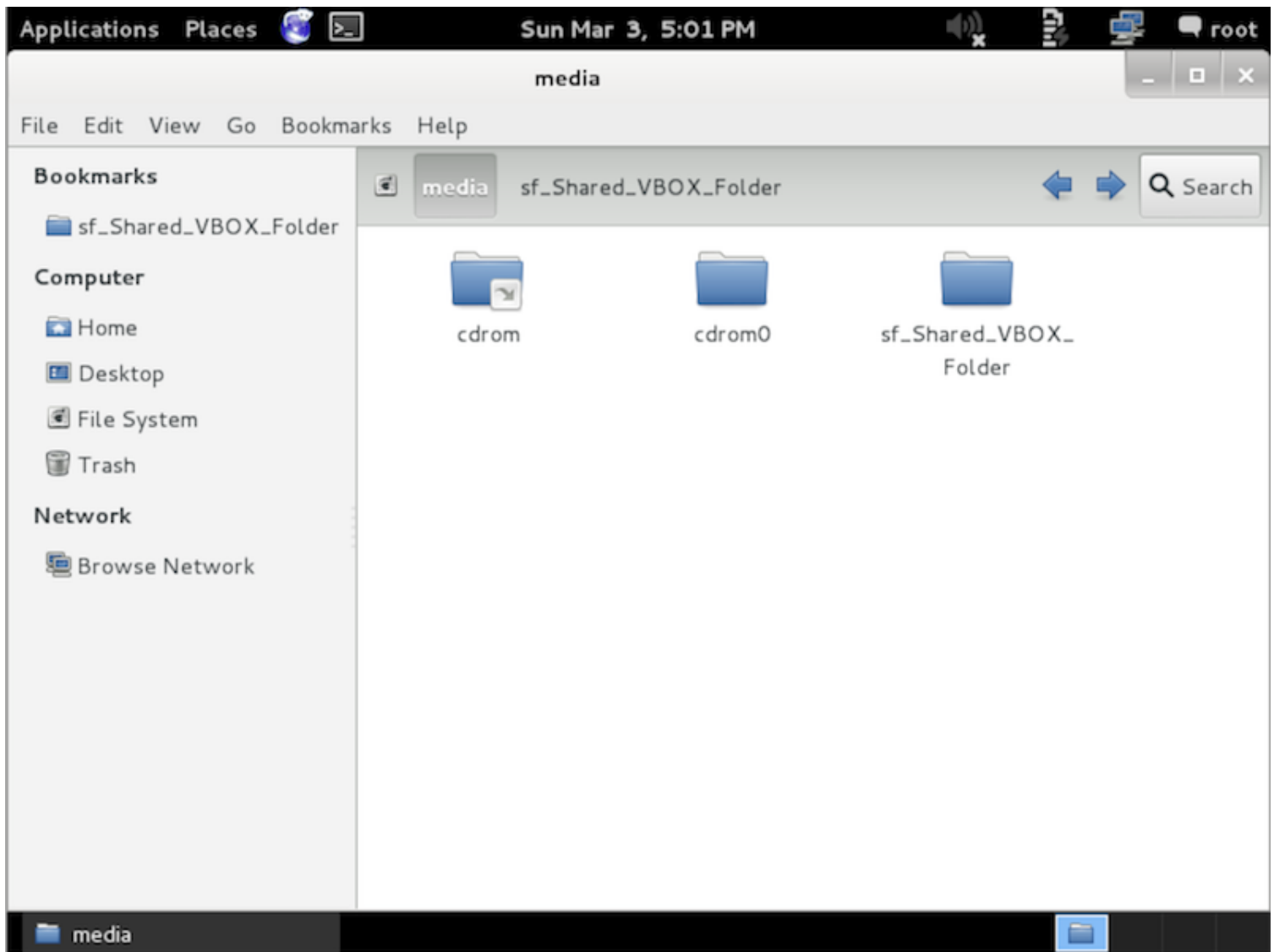
From the VirtualBox Manager, select your Kali Linux VM instance and click on the 'Shared Folders' link in the right window pane. This will launch a pop up window for adding shared folders. Within this window click the icon to add a folder.

In the Folder Path text box, provide the path to the folder you would like to share, or click the drop-down arrow to browse your host system for the path. Select the check boxes that allow for 'Auto-mount' and 'Make

Permanent' and click the OK button both times when prompted.



Your shared folders will now be available in the media directory. You can create a bookmark or link for easier access to the directory.



Starting Metasploit Framework

In keeping with the [Kali Linux Network Services Policy](#), there are no network services, including database services, running on boot so there are a couple of steps that need to be taken in order to get [Metasploit](#) up and running with database support.

Start the Kali PostgreSQL Service

Metasploit uses [PostgreSQL](#) as its database so it needs to be launched first.

```
service postgresql start
```

You can verify that PostgreSQL is running by checking the output of **ss -ant** and making sure that port 5432 is listening.

```
State Recv-Q Send-Q Local Address:Port Peer Address:Port
LISTEN 0 128 :::22 :::*
LISTEN 0 128 *:22 *:*
LISTEN 0 128 127.0.0.1:5432 *:*
LISTEN 0 128 ::1:5432 :::*
```

Start the Kali Metasploit Service

With PostgreSQL up and running, we next need to launch the metasploit service. The first time the service is launched, it will create a msf3 database user and a database called msf3. The service will also launch the Metasploit RPC and Web servers it requires.

```
service metasploit start
```

Launch msfconsole in Kali

Now that the PostgreSQL and Metasploit services are running, you can launch **msfconsole** and verify database connectivity with the **db_status** command as shown below.

```
msfconsole
```

```
msf > db_status  
[*] postgresql connected to msf3  
msf >
```

Configure Metasploit to Launch on Startup

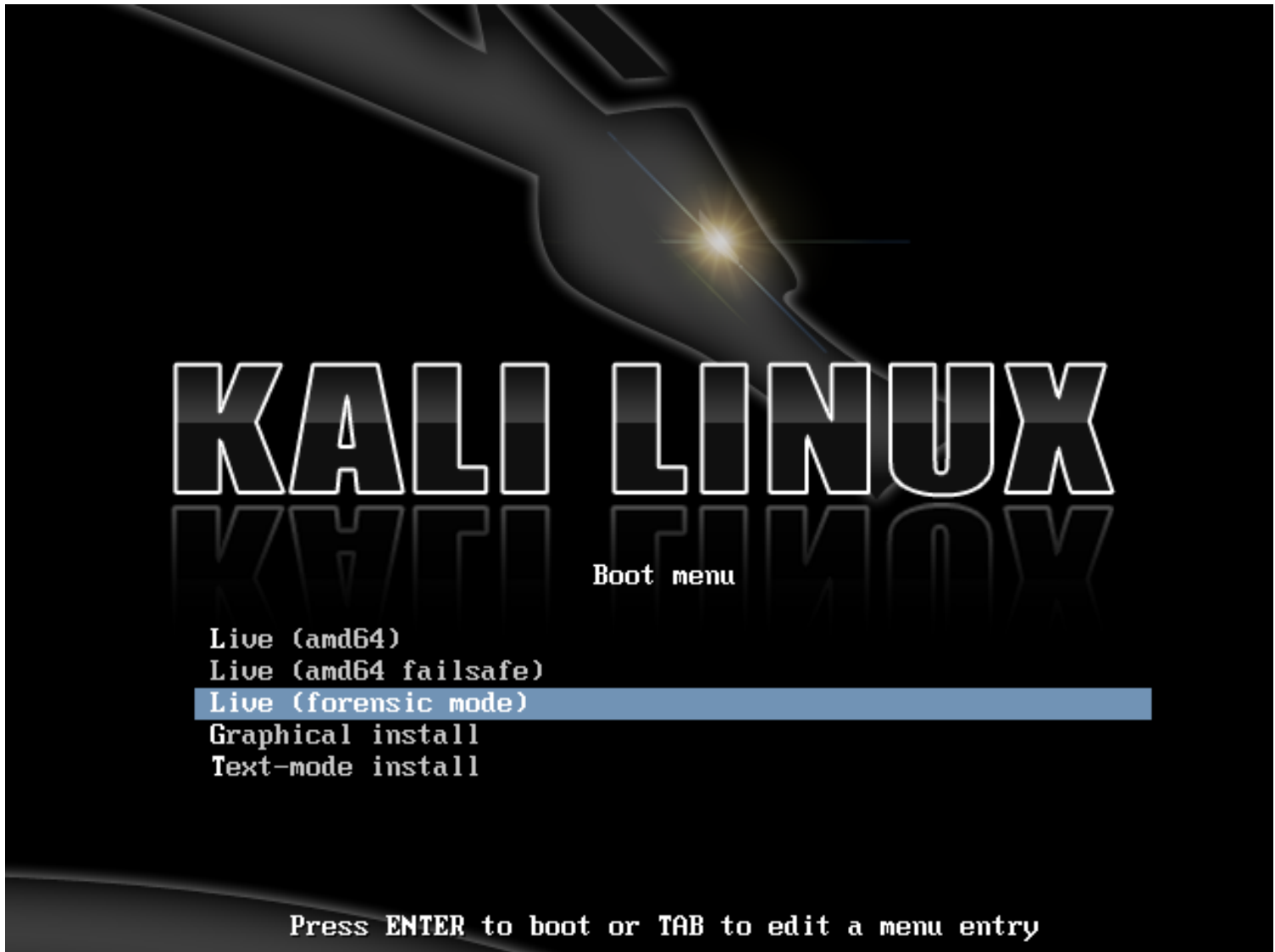
If you would prefer to have PostgreSQL and Metasploit launch at startup, you can use **update-rc.d** to enable the services as follows.

```
update-rc.d postgresql enable
```

```
update-rc.d metasploit enable
```


Kali Linux Forensics Mode

BackTrack Linux introduced a “Forensic Boot” option to the operating system that continued on through BackTrack 5 and now exists in Kali Linux. The “Forensics Boot” option has proven to be very popular due to the widespread availability of our operating system. Many people have Kali ISOs laying around and when a forensic need comes up, it is quick and easy to put Kali Linux to the job. Pre-loaded with the most popular open source forensic software, Kali is a handy tool when you need to do some open source forensic work.



When booted into the forensic boot mode, there are a few very important changes that are made.

1. First off, the internal hard disk is not touched. This means that if there is a swap partition it will not be used and no internal disk will be auto mounted. To verify this, we took a standard system and removed the hard drive. Attaching this to a commercial forensic package we took a hash of the drive. We then re-

attached the drive to the computer and booted up off of Kali in forensic boot mode. After using Kali for a period of time, we then shut the system down, removed the hard drive, and took the hash again. These hashes matched, indicating that at no point was anything changed on the drive at all.

2. The other, just as important, change that was made was we disabled the auto mount of any removable media. So thumb drives, CDs, and so on will not be auto-mounted when inserted. The idea behind all of this is simple: Nothing should happen to any media without direct user action. Anything that you do as a user is on you.

If you are interested in using Kali for real world forensics of any type, we recommend that you don't just take our word for any of this. All forensic tools should always be validated to ensure that you know how they will behave in any circumstance that you may place them.

And finally, as Kali is focused on having the best collection of open source penetration testing tools available, it is possible that we may have missed your favorite open source forensic tool. If so, [let us know!](#) We are always on the lookout of high quality open source tools that we can add to Kali to make it even better.

VMware Tools in a Kali Guest

Should you decide to create your own VMware installation of Kali Linux rather than using our pre-made VMware images, you will need to follow the instructions below in order to successfully install VMware Tools in your Kali installation. You can opt to install either **open-vm-tools**, or the native **VMWare tools**.

Installing open-vm-Tools

This is probably the easiest way to get “VMWare tools” functionality inside a kali VMWare guest.

```
apt-get install open-vm-tools
```

Installing VMware Tools in Kali

If open-vm-tools does not work for you, or if you prefer using native VMWare tools, begin by installing some packages that are required by the VMware Tools installer:

```
echo cups enabled >> /usr/sbin/update-rc.d
echo vmware-tools enabled >> /usr/sbin/update-rc.d

apt-get install gcc make linux-headers-$(uname -r)
ln -s /usr/src/linux-headers-$(uname -r)/include/generated/uapi/linux/version.h /usr/src/linux-headers-$(uname -r)/include/linux/
```

Next, mount the VMware tools ISO by clicking “Install VMware Tools” from the appropriate menu. Once the VMware Tools ISO has been attached to the virtual machine, we mount the drive and copy the VMware Tools installer to /tmp/.

```
mkdir /mnt/vmware
```

```
mount /dev/cdrom /mnt/vmware/  
cp -rf /mnt/vmware/VMwareTools* /tmp/
```

Finally, change directory to /tmp/, extract the tarball and start the installer:

```
cd /tmp/  
tar xzpf VMwareTools-*.tar.gz  
cd vmware-tools-distrib/  
./vmware-tools-install.pl
```

Follow the prompts for the VMware Tools installation and you are done.

Slow Mouse Movement in VMware

If your mouse movement is slow and sluggish in a Kali Linux VMware guest, try installing the **xserver-xorg-input-vmmouse** package in the Kali guest.

```
apt-get install xserver-xorg-input-vmmouse  
reboot
```

VMware Tools Won't Compile!

This is an unfortunate reality that has often plagued us, as Kali Linux uses a bleeding edge kernel which is not always supported by VMware. On occasion, it might be required to search for “upstream compatibility VMware Tools patches” from the [VMware community](#).

Known Issues

As of March 2nd, 2013, VMware tools will compile with kernel 3.7, barring the shared folder module. [Patches](#) exist to fix this issue..

06. Kali Linux ARM Architecture

Install Kali ARM on an EfikaMX



The EfikaMX is a low end, low cost ARM computer. Despite its less-than-stellar specifications, its affordability makes it an excellent option for a tiny Linux system.

Stock Kali on EfikaMX - Easy Version

If all you want to do is to install Kali on your EfikaMX, follow these instructions:

1. Get a nice fast 8 GB (or more) SD card. Class 10 cards are highly recommended.
2. Download the Kali Linux EfikaMX image from our [downloads](#) area.
3. Use the **dd** utility to image this file to your SD card. In our example, we assume the storage device is located at `/dev/sdb`. **Change this as needed.**

Alert! This process will wipe out your SD card. If you choose the wrong storage device, you may wipe out your computers hard disk.

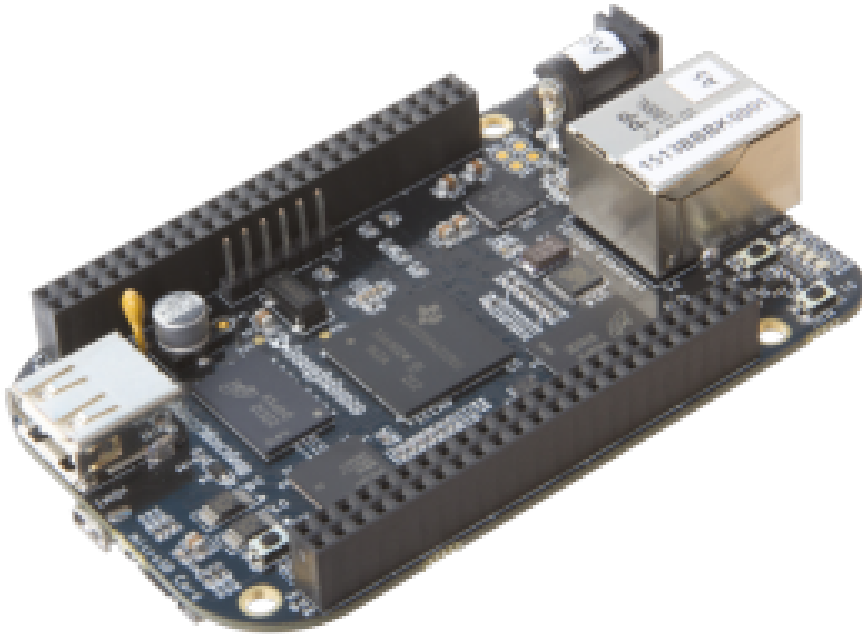
```
root@kali:~ dd if=kali-1.0.3-efikamx.img of=/dev/sdb bs=512k
```

This process can take a while depending on your USB storage device speed and image size. Once the dd operation is complete, boot up your EfikaMX with the SD card plugged in. You will be able to log in to Kali (root / toor) and **startx**. That's it, you're done!

Kali on EfikaMX - Long Version

If you are a developer and want to tinker with the Kali EfikaMX image, including changing the kernel configuration and generally being adventurous, check out our [Custom EfikaMX Image](#) article.

Kali ARM on a Beaglebone Black



The Beaglebone Black is a low end, low cost ARM computer. Despite its less-than-stellar specifications, its affordability makes it an excellent option for a tiny Linux system.

Stock Kali on Beaglebone Black - Easy Version

If all you want to do is to install Kali on your Beaglebone Black, follow these instructions:

1. Get a nice fast 8 GB (or more) SD card. Class 10 cards are highly recommended.
2. Download the Kali Linux Beaglebone Black image from our [downloads](#) area.
3. Use the **dd** utility to image this file to your SD card. In our example, we assume the storage device is located at `/dev/sdb`. **Change this as needed.**

Alert! This process will wipe out your SD card. If you choose the wrong storage device, you may wipe out your computers hard disk.

```
root@kali:~ dd if=kali-bbb.img of=/dev/sdb bs=512k
```


This process can take a while depending on your USB storage device speed and image size. Once the dd operation is complete, boot up your Beaglebone Black with the SD card plugged in. You will be able to log in to Kali (root / toor) and **startx**. That's it, you're done!

Kali on Beaglebone Black - Long Version

If you are a developer and want to tinker with the Kali Beaglebone Black image, including changing the kernel configuration and generally being adventurous, check out our [Custom Beaglebone Black Image](#) article.

Install Kali ARM on a CuBox



The CuBox is a low end, low cost ARM computer. Despite its less-than-stellar specifications, its affordability makes it an excellent option for a tiny Linux system and it can do far more than act as a media PC.

Stock Kali on CuBox - Easy Version

If all you want to do is to install Kali on your CuBox, follow these instructions:

1. Get a nice fast 8 GB (or more) SD card. Class 10 cards are highly recommended.
2. Download the Kali Linux CuBox image from our [downloads](#) area.
3. Use the **dd** utility to image this file to your SD card. In our example, we assume the storage device is located at `/dev/sdb`. **Change this as needed.**

Alert! This process will wipe out your SD card. If you choose the wrong storage device, you may wipe out your computers hard disk.

```
root@kali:~ dd if=kali-1.0.3-cubox.img of=/dev/sdb bs=512k
```

This process can take a while depending on your USB storage device speed and image size. Once the dd operation is complete, boot up your CuBox with the SD card plugged in. You will be able to log in to Kali (root / toor) and **startx**. That's it, you're done!

Kali on CuBox - Long Version

If you are a developer and want to tinker with the Kali CuBox image, including changing the kernel configuration and generally being adventurous, check out our [Custom CuBox Image](#) article.

Kali Linux on Galaxy Note 10.1



The Samsung Galaxy Note 10.1 is a 10.1-inch tablet computer designed, developed, and marketed by Samsung. The tablet incorporates a 1.4 GHz quad-core Exynos processor and 2 GB of RAM. The touch screen works surprisingly well with Kali as well as the wireless card, however Bluetooth and audio are not yet functional on this image.

Stock Kali on Galaxy Note 10.1 - Easy Version

If all you want to do is to install Kali on your Galaxy Note 10.1, follow these instructions:

1. You'll need **at least 7 GB free** on your internal SD card for our image.
2. Root your Samsung Galaxy Note 10.1 if you have not already done so.
3. Download the **Kali Linux Galaxy Note 10.1** image from our [downloads](#) area.
4. Rename the downloaded Kali image to **linux.img** and copy it to `/storage/sdcard0`.

5. Download our recovery.img file from [here](#) and copy it to /storage/sdcard0.
6. Get root on your Galaxy Note 10.1, change /storage/sdcard0, and backup your recovery partition:

```
dd if=/dev/block/mmcblk0p6 of=recovery.img_orig
```

1. **dd** the downloaded recovery.img image to the recovery partition:

Alert! This process will overwrite your recovery partition. Please make sure you know what you are doing. You may brick your device if you fumble this.

```
dd if=recovery.img of=/dev/block/mmcblk0p6
```

1. Reboot your Galaxy Note 10.1 into recovery mode. You can do this by **turning it off**, then press and hold both the **power button** and the **volume up** button. Once you see the “Samsung Galaxy Note 10.1” text appear, **release the power button but keep pressing the volume up button** . This should boot you into Kali and auto-login into Gnome. The root password is “changeme” (without the quotes!)
2. Open the onscreen keyboard by going to : **Applications -> Universal Access -> Florence Virtual Keyboard**.
3. Wireless works but seems to skip the scanning of networks without some massaging. **If the Gnome Network Manager shows no wireless networks** , simply add your wireless network as a “hidden” one and you should get connected as usual.
4. You can modify, debug, and explore our image easily from within your Galaxy Note, using a wonderful Android App called [Linux Deploy](#).

Install Kali Samsung Chromebook



Samsung ARM Chromebook

The Samsung ARM Chromebook is an ultraportable laptop. It was quite a challenge, but we have a Kali image that runs great on the Chromebook.

Our Kali Chromebook image contains two boot partitions - one holds a kernel hardcoded to boot from SD, and the other holds a kernel hardcoded to boot from USB. Depending on your USB storage media type, make sure to mark the relevant boot partition with higher priority after you dd the image to your USB device, as instructed in the last stages of this guide.

Kali on Chromebook - User Instructions

If all you want to do is install Kali on your Samsung ARM Chromebook, follow these instructions:

1. Get a nice fast 8 GB SD card or USB stick.
2. Put your Chromebook in developer mode.
3. Download the Kali Samsung ARM Chromebook image from our [downloads](#) area.
4. Use the **dd** utility to image this file to your SD /USB device. In our example, we use a USB stick which is located at `/dev/sdb`. **Change this as needed.**

Alert! This process will wipe out your SD card. If you choose the wrong storage device, you may wipe out

your computers hard disk.

```
dd if=kali-chromebook.img of=/dev/sdb bs=512k
```

This process can take awhile depending on your USB storage device speed and image size.

This is the point where you need to mark either boot partition 1 or 2 to have higher priority. The number with the higher priority will boot first. The example below will give priority 10 to the first partition (-i), and will thus boot successfully from a SD card.

```
cgpt repair /dev/sdb
cgpt add -i 1 -S 1 -T 5 -P 10 -I KERN-A /dev/sdb
cgpt add -i 2 -S 1 -T 5 -P 5 -I KERN-B /dev/sdb
```

To see your partition list and order, use the command **cgpt show**.

```
root@kali:~# cgpt show /dev/sdb
start size part contents
0 1 PMBR
1 1 Pri GPT header
2 32 Pri GPT table
8192 32768 1 Label: "KERN-A"
Type: ChromeOS kernel
UUID: 63AD6EC9-AD94-4B42-80E4-798BBE6BE46C
Attr: priority=10 tries=5 successful=1
40960 32768 2 Label: "KERN-B"
Type: ChromeOS kernel
```

```
UUID: 37CE46C9-0A7A-4994-80FC-9C0FFCB4FDC1
Attr: priority=5 tries=5 successful=1
73728 3832490 3 Label: "Linux filesystem"
Type: 0FC63DAF-8483-4772-8E79-3D69D8477DE4
UUID: E9E67EE1-C02E-481C-BA3F-18E721515DBB
125045391 32 Sec GPT table
125045423 1 Sec GPT header
root@kali:~#
```

Once the dd operation is complete, boot up the Chromebook with the SD / USB plugged in (NOT IN THE BLUE USB PORT!). At the developer boot prompt, hit CTRL+ALT+U, which should boot you into Kali Linux. Log in to Kali (root / toor) and **startx**. That's it, you're done!

Kali on Samsung Chromebook - Developer Instructions

If you are a developer and want to tinker with the Kali Samsung Chromebook image, including changing the kernel configuration and generally being adventurous, check out our [Custom Chromebook Kernel / Image](#) article.

Install Kali ARM on MK/SS808



SS808 ARM Devices (rk3306)

The SainSmart SS808 is a **rockchip**-based ARM device that comes in various forms and flavors. It has a dual-core 1.6 GHz A9 processor with 1 GB of RAM and runs Kali very well.

Stock Kali on SS808 - Easy Version

If all you want to do is to install Kali on your SS808, follow instructions below:

1. Get a nice fast 8 GB (or more) microSD card. Class 10 cards are highly recommended.
2. Download the Kali Linux SS808 image from our [downloads](#) area.
3. Use the **dd** utility to image this file to your microSD card. In our example, we assume the storage device is located at /dev/sdb and are using an SS808 image. **Change this as needed.**
4. Download the [MK808-Finless-1-6-Custom-ROM](#) to a Windows machine and extract the zip file.
5. Read the README file of the MK808 Finless ROM tool, then install the required Windows drivers.
6. Run the Finless ROM Flash Tool and ensure that it says "Found RKAndroid Loader Rock USB" at the bottom. Deselect kernel.img and recovery.img from the list, and flash the device.
7. Next overwrite both kernel.img and recovery.img in the Finless ROM directory with the kali "kernel.img".

8. In the Finless ROM tool, make sure only “kernel.img” and “recovery.img” are selected, and flash your device again.
9. Insert your microSD card in the SS808 and boot it up.

Alert! This process will wipe out your SD card! If you choose the wrong storage device, you may wipe out your computers hard disk.

```
dd if=kali-SS808.img of=/dev/sdb bs=1M
```

This process can take a while depending on your USB storage device speed and image size. Once the dd operation is done, boot up your SS808, with the microSD card plugged in. Log in to Kali (root / toor) and **startx**. That's it, you're done!

Kali on SS808 - Long version

If you are a developer and want to tinker with the Kali SS808 image, including changing the kernel configuration and generally being adventurous, check out our [Custom MK/SS808 Image](#) article.

Install Kali ARM on ODROID U2



Odroid U2 / X2

The ODROID U2 is a tricky piece of hardware as console output is not a given. Ideally, when purchasing an ODROID, you should also get a USB UART cable, used for serial debugging of the boot process. Saying this, these machines are (at this time) some of the most impressive in terms of size, horsepower and memory availability.

Kali on ODROID U2 - User Instructions

If all you want to do is to install Kali on your awesome ODROID, follow these instructions:

1. Get a nice fast 8 GB (and above) microSD. Class 10 cards are highly recommended.
2. Download the Kali Linux ODROID U2 image from our [downloads](#) area.
3. Use the **dd** utility to image this file to your microSD card. In our example, we assume the storage device is located at `/dev/sdb`. **Change this as needed.**

Alert! This process will wipe out your SD card. If you choose the wrong storage device, you may wipe out your computers hard disk.

```
dd if=kali-ordoidu2.img of=/dev/sdb bs=1M
```

This process can take a while depending on your USB storage device speed and image size. Once the dd operation is done, boot up the Odroid with the microSD plugged in. You should be welcomed with a Gnome login screen - (root / toor). That's it, you're done!

Troubleshooting

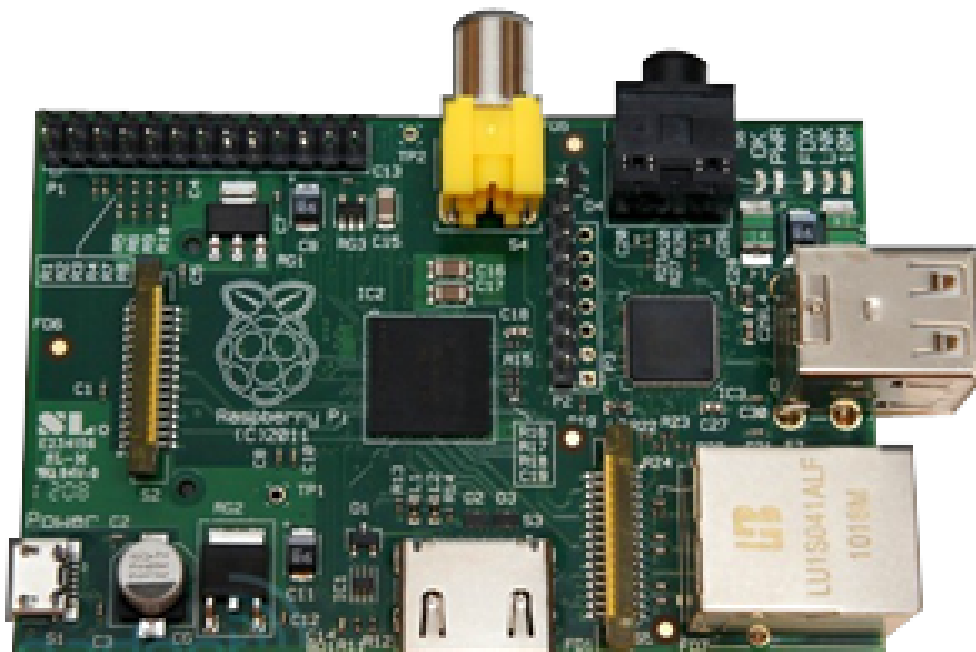
To troubleshoot the Odroid boot process, you will need to connect a UART serial cable to the Odroid. Once the cable is connected, you can issue the following command to connect to the console:

```
screen /dev/ttySAC1 115200
```

Kali on ODROID U2 - Developer Instructions

If you are a developer and want to tinker with the Kali ODROID image, including changing the kernel configuration and generally being adventurous, check out our [Building a Custom Kali ODROID Image](#) article.

Install Kali ARM on a Raspberry Pi



Raspberry Pi

The Raspberry Pi is a low end, low cost ARM computer. Despite its less-than-stellar specifications, its affordability makes it an excellent option for a tiny Linux system and it can do far more than act as a media PC.

Stock Kali on Raspberry Pi - Easy Version

If all you want to do is to install Kali on your Raspberry Pi, follow these instructions:

1. Get a nice fast 8 GB (or more) SD card. Class 10 cards are highly recommended.
2. Download the Kali Linux Raspberry Pi image from our [downloads](#) area.
3. Use the **dd** utility to image this file to your SD card. In our example, we assume the storage device is located at `/dev/sdb`. **Change this as needed.**

Alert! This process will wipe out your SD card. If you choose the wrong storage device, you may wipe out your computers hard disk.

```
root@kali:~ dd if=kali-pi.img of=/dev/sdb bs=512k
```

This process can take a while depending on your USB storage device speed and image size. Once the dd operation is complete, boot up your Raspberry Pi with the SD card plugged in. You will be able to log in to Kali (root / toor) and **startx**. That's it, you're done!

Kali on Raspberry Pi - Long Version

If you are a developer and want to tinker with the Kali Raspberry Pi image, including changing the kernel configuration and generally being adventurous, check out our [Custom Raspberry Pi Image](#) article.

Preparing a Kali Linux ARM chroot

Although you can [download Kali ARM images](#) from our Download area, some prefer building their own updated bootstrapped Kali rootfs. The following procedure shows an example of building a Kali**armhf** rootfs. Change to **armel** if needed.

Install Required Tools and Dependencies

```
apt-get install debootstrap qemu-user-static
```

Define Architecture and Custom Packages

This is where you define some environment variables for your required ARM architecture (armel vs armhf) and list the packages to be installed in your image. These will be used throughout this article, so make sure to modify them to your needs.

```
export packages="xfce4 kali-menu wpasupplicant kali-defaults initramfs-tools uboot-mkimage nmap  
openssh-server"  
export architecture="armhf"
```

Build the Kali rootfs

We create a standard directory structure and bootstrap ARM rootfs from the Kali Linux repositories. We then copy over **qemu-arm-static** from our host machine into the rootfs in order to initiate the 2nd stage chroot.

```
cd ~  
mkdir -p arm-stuff  
cd arm-stuff/  
mkdir -p kernel  
mkdir -p rootfs  
cd rootfs
```

```
debootstrap --foreign --arch $architecture kali kali-$architecture http://archive.kali.org/kali
cp /usr/bin/qemu-arm-static kali-$architecture/usr/bin/
```

2nd Stage chroot

This is where we configure base image settings such as keymaps, repositories, default network interface behavior (change if needed), etc.

```
cd ~/arm-stuff/rootfs
LANG=C chroot kali-$architecture /debootstrap/debootstrap --second-stage

cat << EOF > kali-$architecture/etc/apt/sources.list
deb http://http.kali.org/kali kali main contrib non-free
deb http://security.kali.org/kali-security kali/updates main contrib non-free
EOF

echo "kali" > kali-$architecture/etc/hostname

cat << EOF > kali-$architecture/etc/network/interfaces
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet dhcp
EOF

cat << EOF > kali-$architecture/etc/resolv.conf
nameserver 8.8.8.8
EOF
```

3rd Stage chroot

This is where your customization comes in. Your \$packages are installed and a default “toor” root password is set as well as other configuration changes and fixes.


```
export MALLOC_CHECK_=0 # workaround for LP: #520465
export LC_ALL=C
export DEBIAN_FRONTEND=noninteractive

mount -t proc proc kali-$architecture/proc
mount -o bind /dev/ kali-$architecture/dev/
mount -o bind /dev/pts kali-$architecture/dev/pts

cat << EOF > kali-$architecture/debconf.set
console-common console-data/keymap/policy select Select keymap from full list
console-common console-data/keymap/full select en-latin1-nodeadkeys
EOF

cat << EOF > kali-$architecture/third-stage
#!/bin/bash
dpkg-divert --add --local --divert /usr/sbin/invoke-rc.d.chroot --rename /usr/sbin/invoke-rc.d
cp /bin/true /usr/sbin/invoke-rc.d

apt-get update
apt-get install locales-all
#locale-gen en_US.UTF-8

debconf-set-selections /debconf.set
rm -f /debconf.set
apt-get update
apt-get -y install git-core binutils ca-certificates initramfs-tools uboot-mkimage
apt-get -y install locales console-common less nano git
echo "root:toor" | chpasswd
sed -i -e 's/KERNEL\!=="eth\*"|KERNEL\!=="' /lib/udev/rules.d/75-persistent-net-generator.rules
rm -f /etc/udev/rules.d/70-persistent-net.rules
apt-get --yes --force-yes install $packages

rm -f /usr/sbin/invoke-rc.d
dpkg-divert --remove --rename /usr/sbin/invoke-rc.d

rm -f /third-stage
EOF

chmod +x kali-$architecture/third-stage
LANG=C chroot kali-$architecture /third-stage
```

Manual Configuration Within the chroot

If needed, you can perform any final modifications in your rootfs environment by manually chrooting into it and making any necessary last changes.

```
LANG=C chroot kali-$architecture
{make additional changes within the chroot}
exit
```

Cleanup

Lastly, we run a cleanup script in our chroot to free up space used by cached files and run any other cleanup jobs we may require:

```
cat << EOF > kali-$architecture/cleanup
#!/bin/bash
rm -rf /root/.bash_history
apt-get update
apt-get clean
rm -f cleanup
EOF

chmod +x kali-$architecture/cleanup
LANG=C chroot kali-$architecture /cleanup

umount kali-$architecture/proc
umount kali-$architecture/dev/pts
umount kali-$architecture/dev/

cd ..
```

Congratulations! Your custom Kali ARM rootfs is located in the `kali-$architecture` directory. You can now tar up this directory or copy it to an image file for further work.

07. Kali Linux Development

Custom EfikaMX Image

The following document describes our own method of creating a **custom Kali Linux EfikaMX ARM image** and is targeted at developers. If you would like to install a pre-made Kali image, check out our [Install Kali on an EfikaMX](#) article.

01. Create a Kali rootfs

Build a [Kali rootfs](#) as described in our Kali documentation, using an **armhf** architecture. By the end of this process, you should have a populated rootfs directory in **~/arm-stuff/rootfs/kali-armhf**.

02. Create the Image File

Next, we create the physical image file, which will hold our EfikaMX rootfs and boot images.

```
apt-get install kpartx xz-utils sharutils
cd ~
mkdir -p arm-stuff
cd arm-stuff/
mkdir -p images
cd images
dd if=/dev/zero of=kali-custom-efikamx.img bs=1MB count=7000
```

03. Partition and Mount the Image File

```
parted kali-custom-efikamx.img --script -- mklable msdos
parted kali-custom-efikamx.img --script -- mkpart primary ext2 4096s 266239s
parted kali-custom-efikamx.img --script -- mkpart primary ext4 266240s 100%
```

```
loopdevice=`losetup -f --show kali-custom-efikamx.img`
```

```
device=`kpartx -va $loopdevice| sed -E 's/.*(loop[0-9])p.*\1/g' | head -1`
device="/dev/mapper/${device}"
bootp=${device}p1
rootp=${device}p2

mkfs.ext2 $bootp
mkfs.ext4 $rootp
mkdir -p boot
mkdir -p root
mount $bootp boot
mount $rootp root
```

04. Copy and Modify the Kali rootfs

```
rsync -HPavz /root/arm-stuff/rootfs/kali-armhf/ root
echo nameserver 8.8.8.8 > root/etc/resolv.conf
sed 's/0-1/0//g' root/etc/init.d/udev
```

05. Compile the EfikaMX Kernel and Modules

If you're not using ARM hardware as the development environment, you will need to set up an [ARM cross-compilation environment](#) to build an ARM kernel and modules. Once that's done, proceed with the following instructions.

```
cd ~/arm-stuff
mkdir -p kernel
cd kernel
git clone --depth 1 https://github.com/genesi/linux-legacy.git
cd linux-legacy
export ARCH=arm
export CROSS_COMPILE=~/.arm-stuff/kernel/toolchains/arm-eabi-linaro-4.6.2/bin/arm-eabi-
make efikamx_defconfig
# configure your kernel !
make menuconfig
```

```
make -j$(cat /proc/cpuinfo|grep processor|wc -l)
make modules_install INSTALL_MOD_PATH=~/.arm-stuff/images/root
make ulmage
cp arch/arm/boot/ulmage ~/.arm-stuff/images/boot

cat << EOF > ~/.arm-stuff/images/boot/boot.script
setenv ramdisk ulnitrd;
setenv kernel ulmage;
setenv bootargs console=tty1 root=/dev/mmcbk0p2 rootwait rootfstype=ext4 rw quiet;
${loadcmd} ${ramdiskaddr} ${ramdisk};
if imi ${ramdiskaddr}; then; else
setenv bootargs ${bootargs} noinitrd;
setenv ramdiskaddr "";
fi;
${loadcmd} ${kerneladdr} ${kernel}
if imi ${kerneladdr}; then
bootm ${kerneladdr} ${ramdiskaddr}
fi;
EOF

mkimage -A arm -T script -C none -n "Boot.scr for EfikaMX" -d ~/.arm-stuff/images/boot/boot.script
~/.arm-stuff/images/boot/boot.scr
```

```
umount $bootp
umount $rootp
kpartx -dv $loopdevice
losetup -d $loopdevice
```

Use the **dd** utility to image this file to your SD card. In our example, we assume the storage device is located at `/dev/sdb`. **Change this as needed.**

```
dd if=kali-custom-efikamx.img of=/dev/sdb bs=1M
```

Once the dd operation is complete, unmount and eject the SD card and boot your EfikaMX into Kali Linux

Custom Beaglebone Black Image

The following document describes our own method of creating a **custom Kali Linux Beaglebone Black ARM image** and is targeted at developers. If you would like to install a pre-made Kali image, check out our [Install Kali on Beaglebone Black](#) article.

01. Create a Kali rootfs

Build a [Kali rootfs](#) as described in our Kali documentation, using an **armhf** architecture. By the end of this process, you should have a populated rootfs directory in **~/arm-stuff/rootfs/kali-armhf**.

02. Create the Image File

Next, we create the physical image file, which will hold our Beaglebone Black rootfs and boot images.

```
apt-get install kpartx xz-utils sharutils
cd ~
mkdir -p arm-stuff
cd arm-stuff/
mkdir -p images
cd images
dd if=/dev/zero of=kali-custom-bbb.img bs=1MB count=7000
```

03. Partition and Mount the Image File

```
parted --script kali-custom-bbb.img mklabel msdos
fdisk kali-custom-bbb.img << __EOF__
n
p
1

+64M
t
e
p
```



```
w
__EOF__
parted --script kali-custom-bbb.img set 1 boot on
fdisk kali-custom-bbb.img << __EOF__
n
p
2

w
__EOF__
```

```
loopdevice=`losetup -f --show kali-custom-bbb.img`
device=`kpartx -va $loopdevice| sed -E 's/.*(loop[0-9])p.*\1/g' | head -1`
device="/dev/mapper/${device}"
bootp=${device}p1
rootp=${device}p2

mkfs.vfat -F 16 $bootp -n boot
mkfs.ext4 $rootp -L kaliroot
mkdir -p boot
mkdir -p root
mount $bootp boot
mount $rootp root
```

04. Copy and Modify the Kali rootfs

```
rsync -HPavz /root/arm-stuff/rootfs/kali-armhf/ root
echo nameserver 8.8.8.8 > root/etc/resolv.conf
```

05. Compile the Beaglebone Black Kernel and Modules

If you're not using ARM hardware as the development environment, you will need to set up an [ARM cross-](#)

[compilation environment](#) to build an ARM kernel and modules. Once that's done, proceed with the following instructions.

```
cd ~/arm-stuff
wget
https://launchpad.net/linaro-toolchain-binaries/trunk/2013.03/+download/
gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux.tar.bz2
tar xjf gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux.tar.bz2
export CC=`pwd`/gcc-linaro-arm-linux-gnueabi-4.7-2013.03-20130313_linux/bin/arm-linux-gnueabi-

git clone git://git.denx.de/u-boot.git
cd u-boot/
git checkout v2013.04 -b beaglebone-black
wget
https://raw.githubusercontent.com/eewiki/u-boot-patches/master/v2013.04/0001-am335x_evm-uEnv.txt-bootz-n-
fixes.patch
patch -p1 < 0001-am335x_evm-uEnv.txt-bootz-n-fixes.patch
make ARCH=arm CROSS_COMPILE=${CC} distclean
make ARCH=arm CROSS_COMPILE=${CC} am335x_evm_config
make ARCH=arm CROSS_COMPILE=${CC}
cd ..

mkdir -p kernel
cd kernel
git clone git://github.com/RobertCNelson/linux-dev.git
cd linux-dev/
git checkout origin/am33x-v3.8 -b tmp
./build_kernel.sh
mkdir -p ../patches
wget http://patches.aircrack-ng.org/mac80211.compat08082009.wl_frag+ack_v1.patch -O
../patches/mac80211.patch
cd KERNEL
patch -p1 --no-backup-if-mismatch < ../patches/mac80211.patch
cd ..
./tools/rebuild.sh
cd ..

cat << EOF > boot/uEnv.txt
mmcroot=/dev/mmcblk0p2 ro
```

```
mmcrootfstype=ext4 rootwait fixrtc
uenvcmd=run loaduimage; run loadfdt; run mmcargs; bootz 0x80200000 - 0x80F80000
EOF
```

```
cp -v kernel/linux-dev/deploy/3.8.13-bone20.zImage boot/zImage
mkdir -p boot/dtbs
tar -xovf kernel/linux-dev/deploy/3.8.13-bone20-dtbs.tar.gz -C boot/dtbs/

tar -xovf kernel/linux-dev/deploy/3.8.13-bone20-modules.tar.gz -C root/
tar -xovf kernel/linux-dev/deploy/3.8.13-bone20-firmware.tar.gz -C root/lib/firmware/
```

```
cat << EOF > root/etc/fstab
/dev/mmcblk0p2 / auto errors=remount-ro 0 1
/dev/mmcblk0p1 /boot/uboot auto defaults 0 0
EOF
```

```
umount $rootp
kpartx -dv $loopdevice
losetup -d $loopdevice
```

Use the **dd** utility to image this file to your SD card. In our example, we assume the storage device is located at `/dev/sdb`. **Change this as needed.**

```
dd if=kali-custom-bbb.img of=/dev/sdb bs=1M
```

Once the `dd` operation is complete, unmount and eject the SD card and boot your Beaglebone Black into Kali Linux. When booting you will need to press and hold the “BOOT” button, it’s the one closest to the microSD card.

Custom CuBox Image

The following document describes our own method of creating a **custom Kali Linux CuBox ARM image** and is targeted at developers. If you would like to install a pre-made Kali image, check out our [Install Kali on CuBox](#) article.

01. Create a Kali rootfs

Build a [Kali rootfs](#) as described in our Kali documentation, using an **armhf** architecture. By the end of this process, you should have a populated rootfs directory in **~/arm-stuff/rootfs/kali-armhf**.

02. Create the Image File

Next, we create the physical image file, which will hold our CuBox rootfs and boot images.

```
apt-get install kpartx xz-utils sharutils
cd ~
mkdir -p arm-stuff
cd arm-stuff/
mkdir -p images
cd images
dd if=/dev/zero of=kali-custom-cubox.img bs=1MB count=7000
```

03. Partition and Mount the Image File

```
parted kali-custom-cubox.img --script -- mklabel msdos
parted kali-custom-cubox.img --script -- mkpart primary ext4 0 -1
```

```
loopdevice=`losetup -f --show kali-custom-cubox.img`
device=`kpartx -va $loopdevice | sed -E 's/.*(loop[0-9])p.*/\1/g' | head -1`
device="/dev/mapper/${device}"
rootp=${device}p1
```

```
mkfs.ext4 $rootp
mkdir -p root
mount $rootp root
```

04. Copy and Modify the Kali rootfs

```
rsync -HPavz /root/arm-stuff/rootfs/kali-armhf/ root
echo nameserver 8.8.8.8 > root/etc/resolv.conf
```

05. Compile the CuBox Kernel and Modules

If you're not using ARM hardware as the development environment, you will need to set up an [ARM cross-compilation environment](#) to build an ARM kernel and modules. Once that's done, proceed with the following instructions.

```
cd ~/arm-stuff
mkdir -p kernel
cd kernel
git clone --depth 1 https://github.com/rabeeh/linux.git
cd linux
touch .scmversion
mkdir -p ../patches
wget http://patches.aircrack-ng.org/mac80211.compat08082009.wl_frag+ack_v1.patch -O
../patches/mac80211.patch
patch -p1 --no-backup-if-mismatch < ../patches/mac80211.patch
export ARCH=arm
export CROSS_COMPILE=~/.arm-stuff/kernel/toolchains/arm-eabi-linaro-4.6.2/bin/arm-eabi-
make cubox_defconfig
# configure your kernel !
make menuconfig
make -j$(cat /proc/cpuinfo|grep processor|wc -l)
make modules_install INSTALL_MOD_PATH=~/.arm-stuff/images/root
make ulmage
```

```
cp arch/arm/boot/ulmage ~/arm-stuff/images/root/boot

cat << EOF > ~/arm-stuff/images/root/boot/boot.txt
echo "==" Executing ${directory}${bootscript} on ${device_name} partition ${partition} =="
setenv unit_no 0
setenv root_device ?

if itest.s ${device_name} -eq usb; then
itest.s $root_device -eq ? && ext4ls usb 0:1 /dev && setenv root_device /dev/sda1 && setenv unit_no 0
itest.s $root_device -eq ? && ext4ls usb 1:1 /dev && setenv root_device /dev/sda1 && setenv unit_no 1
fi

if itest.s ${device_name} -eq mmc; then
itest.s $root_device -eq ? && ext4ls mmc 0:2 /dev && setenv root_device /dev/mmcblk0p2
itest.s $root_device -eq ? && ext4ls mmc 0:1 /dev && setenv root_device /dev/mmcblk0p1
fi

if itest.s ${device_name} -eq ide; then
itest.s $root_device -eq ? && ext4ls ide 0:1 /dev && setenv root_device /dev/sda1
fi

if itest.s $root_device -ne ?; then
setenv bootargs "console=ttyS0,115200n8 vmalloc=448M video=dovefb:lcd0:1920x1080-32@60-edid
clcd.lcd0_enable=1 clcd.lcd1_enable=0 root=${root_device} rootfstype=ext4"
setenv loadimage "${fstype}load ${device_name} ${unit_no}:${partition} 0x00200000
${directory}${image_name}"
$loadimage && bootm 0x00200000
echo "!!! Unable to load ${directory}${image_name} from ${device_name} ${unit_no}:${partition} !!!"
exit
fi

echo "!!! Unable to locate root partition on ${device_name} !!!"
EOF

mkimage -A arm -T script -C none -n "Boot.scr for CuBox" -d ~/arm-stuff/images/root/boot/boot.txt
~/arm-stuff/images/root/boot/boot.scr
```

```
umount $rootp  
kpartx -dv $loopdevice  
losetup -d $loopdevice
```

Use the **dd** utility to image this file to your SD card. In our example, we assume the storage device is located at `/dev/sdb`. **Change this as needed.**

```
dd if=kali-custom-cubox.img of=/dev/sdb bs=1M
```

Once the `dd` operation is complete, unmount and eject the SD card and boot your CuBox into Kali Linux

Custom Raspberry Pi Image

The following document describes our own method of creating a **custom Kali Linux Raspberry Pi ARM image** and is targeted at developers. If you would like to install a pre-made Kali image, check out our [Install Kali on Raspberry Pi](#) article.

01. Create a Kali rootfs

Build a [Kali rootfs](#) as described in our Kali documentation, using an **armel** architecture. By the end of this process, you should have a populated rootfs directory in **~/arm-stuff/rootfs/kali-armel**.

02. Create the Image File

Next, we create the physical image file, which will hold our Raspberry Pi rootfs and boot images.

```
apt-get install kpartx xz-utils sharutils
cd ~
mkdir -p arm-stuff
cd arm-stuff/
mkdir -p images
cd images
dd if=/dev/zero of=kali-custom-rpi.img bs=1MB count=7000
```

03. Partition and Mount the Image File

```
parted kali-custom-rpi.img --script -- mklabel msdos
parted kali-custom-rpi.img --script -- mkpart primary fat32 0 64
parted kali-custom-rpi.img --script -- mkpart primary ext4 64 -1
```

```
loopdevice=`losetup -f --show kali-custom-rpi.img`
device=`kpartx -va $loopdevice| sed -E 's/.*(loop[0-9])p.*\1/g' | head -1`
device="/dev/mapper/${device}"
```



```
bootp=${device}p1
rootp=${device}p2

mkfs.vfat $bootp
mkfs.ext4 $rootp
mkdir -p root
mkdir -p boot
mount $rootp root
mount $bootp boot
```

04. Copy and Modify the Kali rootfs

```
rsync -HPavz /root/arm-stuff/rootfs/kali-armel/ root
echo nameserver 8.8.8.8 > root/etc/resolv.conf
```

05. Compile the Raspberry Pi Kernel and Modules

If you're not using ARM hardware as the development environment, you will need to set up an [ARM cross-compilation environment](#) to build an ARM kernel and modules. Once that's done, proceed with the following instructions.

```
cd ~/arm-stuff
mkdir -p kernel
cd kernel
git clone https://github.com/raspberrypi/tools.git
git clone https://github.com/raspberrypi/linux.git raspberrypi
cd raspberrypi
touch .scmversion
export ARCH=arm
export CROSS_COMPILE=~/.arm-stuff/kernel/toolchains/arm-eabi-linaro-4.6.2/bin/arm-eabi-
make bcmrpi_cutdown_defconfig
# configure your kernel !
make menuconfig
make -j$(cat /proc/cpuinfo|grep processor|wc -l)
```

```
make modules_install INSTALL_MOD_PATH=~/.arm-stuff/images/root
cd ../tools/mkimage/
python imagetool-uncompressed.py ../../raspberrypi/arch/arm/boot/Image
```

```
cd ~/.arm-stuff/images
git clone git://github.com/raspberrypi/firmware.git rpi-firmware
cp -rf rpi-firmware/boot/* boot/
rm -rf rpi-firmware

cp ~/.arm-stuff/kernel/tools/mkimage/kernel.img boot/
echo "dwc_otg.lpm_enable=0 console=ttyAMA0,115200 kgdboc=ttyAMA0,115200 console=tty1
root=/dev/mmcblk0p2 rootfstype=ext4 rootwait" > boot/cmdline.txt
```

```
umount $rootp
umount $bootp
kpartx -dv $loopdevice
losetup -d $loopdevice
```

Use the **dd** utility to image this file to your SD card. In our example, we assume the storage device is located at `/dev/sdb`. **Change this as needed.**

```
dd if=kali-pi.img of=/dev/sdb bs=1M
```

Once the `dd` operation is complete, unmount and eject the SD card and boot your Pi into Kali Linux

Custom Chromebook Image

The following document describes our own method of creating a **custom Kali Linux Samsung Chromebook ARM image** and is targeted at developers. If you would like to install a pre-made Kali image, check out our [Install Kali on Samsung Chromebook](#) article.

In this guide, we create an image with two boot partitions - one containing a kernel hard-coded to boot from the SD card and the other containing a kernel hard-coded to boot from USB. Depending on your USB storage media type, make sure to mark the relevant boot partition with higher priority after you dd the image to your USB device as instructed in the last stages of this guide.

01. Create a Kali rootfs

Start by building a [Kali rootfs](#) as described in our Kali documentation, using an **armhf** architecture. By the end of this process, you should have a populated rootfs directory in **~/arm-stuff/rootfs/kali-armhf**.

02. Create the Image File

Next, we create the physical image file that will hold our Chromebook rootfs and boot images.

```
apt-get install kpartx xz-utils gdisk uboot-mkimage u-boot-tools vboot-kernel-utils vboot-utils cgpt
cd ~
mkdir -p arm-stuff
cd arm-stuff/
mkdir -p images
cd images
dd if=/dev/zero of=kali-custom-chrome.img bs=1MB count=7000
```

03. Partition and Mount the Image File

```
parted kali-custom-chrome.img --script -- mklabel msdos
parted kali-custom-chrome.img --script -- mktable gpt
gdisk kali-custom-chrome.img << EOF
x
l
```

```
8192
m
n
1

+16M
7f00
n
2

+16M
7f00
n
3

w
y
EOF
```

```
loopdevice=`losetup -f --show kali-custom-chrome.img`
device=`kpartx -va $loopdevice| sed -E 's/.*(loop[0-9])p.*\1/g' | head -1`
device="/dev/mapper/${device}"
bootp1=${device}p1
bootp2=${device}p2
rootp=${device}p3

mkfs.ext4 $rootp
mkdir -p root
mount $rootp root
```

04. Copy and Modify the Kali rootfs

Copy over the Kali rootfs you bootstrapped earlier using **rsync** to the mounted image.

```
cd ~/arm-stuff/images/
rsync -HPavz ~/arm-stuff/rootfs/kali-armhf/ root

echo nameserver 8.8.8.8 > root/etc/resolv.conf

mkdir -p root/etc/X11/xorg.conf.d/
cat << EOF > root/etc/X11/xorg.conf.d/50-touchpad.conf
Section "InputClass"
Identifier "touchpad"
MatchIsTouchpad "on"
Driver "synaptics"
Option "TapButton1" "1"
Option "TapButton2" "3"
Option "TapButton3" "2"
Option "FingerLow" "15"
Option "FingerHigh" "20"
Option "FingerPress" "256"
EndSection
EOF
```

05. Compile the Samsung Chromium Kernel and Modules

If you're not using ARM hardware as the development environment, you will need to set up an [ARM cross-compilation environment](#) to build an ARM kernel and modules. Once that's done, proceed with the following instructions.

Fetch the Chromium kernel sources and place them in our development tree structure:

```
cd ~/arm-stuff
mkdir -p kernel
cd kernel
git clone http://git.chromium.org/chromiumos/third_party/kernel.git -b chromeos-3.4 chromeos
cd chromeos
```

```
cat << EOF > kernel.its
/dts-v1/;

/ {
description = "Chrome OS kernel image with one or more FDT blobs";
#address-cells = ;
images {
kernel@1{
description = "kernel";
data = /incbin/"arch/arm/boot/zImage");
type = "kernel_noload";
arch = "arm";
os = "linux";
compression = "none";
load = ;
entry = ;
};
fdt@1{
description = "exynos5250-snow.dtb";
data = /incbin/"arch/arm/boot/exynos5250-snow.dtb");
type = "flat_dt";
arch = "arm";
compression = "none";
hash@1{
algo = "sha1";
};
};
};
configurations {
default = "conf@1";
conf@1{
kernel = "kernel@1";
fdt = "fdt@1";
};
};
};
EOF
```

Patch the kernel, in our case, with wireless injection patches.

```
mkdir -p ../patches
wget http://patches.aircrack-ng.org/mac80211.compat08082009.wl_frag+ack_v1.patch -O
../patches/mac80211.patch
wget http://patches.aircrack-ng.org/channel-negative-one-maxim.patch -O ../patches/negative.patch
patch -p1 < ../patches/negative.patch
patch -p1 < ../patches/mac80211.patch
```

Configure, then cross-compile the Chromium kernel as shown below.

```
export ARCH=arm
export CROSS_COMPILE=~/.arm-stuff/kernel/toolchains/arm-eabi-linaro-4.6.2/bin/arm-eabi-

./chromeos/scripts/prepareconfig chromeos-exynos5
# Disable LSM
sed -i 's/CONFIG_SECURITY_CHROMIUMOS=y/# CONFIG_SECURITY_CHROMIUMOS is not set/g' .config
# If cross compiling, do this once:
sed -i 's/if defined(__linux__)/if defined(__linux__) || defined(__KERNEL__) /g' include/drm/drm.h

make menuconfig
make -j$(cat /proc/cpuinfo|grep processor|wc -l)
make dtbs
cp ./scripts/dtc/dtc /usr/bin/
mkimage -f kernel.its kernel.itb
make modules_install INSTALL_MOD_PATH=~/.arm-stuff/images/root/

# copy over firmware. Ideally use the original firmware (/lib/firmware) from the Chromebook.
git clone git://git.kernel.org/pub/scm/linux/kernel/git/dwmw2/linux-firmware.git
cp -rf linux-firmware/* ~/.arm-stuff/images/root/lib/firmware/
rm -rf linux-firmware
```

```
echo "console=tty1 debug verbose root=/dev/mmcblk1p3 rootwait rw rootfstype=ext4" > /tmp/config-sd
echo "console=tty1 debug verbose root=/dev/sda3 rootwait rw rootfstype=ext4" > /tmp/config-usb
```

```
vbutil_kernel --pack /tmp/newkern-sd --keyblock /usr/share/vboot/devkeys/kernel.keyblock --version 1
--signprivate /usr/share/vboot/devkeys/kernel_data_key.vbprivk --config=/tmp/config-sd --vmlinuz kernel.itb
--arch arm
vbutil_kernel --pack /tmp/newkern-usb --keyblock /usr/share/vboot/devkeys/kernel.keyblock --version 1
--signprivate /usr/share/vboot/devkeys/kernel_data_key.vbprivk --config=/tmp/config-usb --vmlinuz
kernel.itb --arch arm
```

06. Prepare the Boot Partition

```
dd if=/tmp/newkern-sd of=$bootp1 # first boot partition for SD
dd if=/tmp/newkern-usb of=$bootp2 # second boot partition for USB
```

```
umount $rootp
```

```
kpartx -dv $loopdevice
losetup -d $loopdevice
```

07. dd the Image and Mark the USB Drive Bootable

```
dd if=kali-custom-chrome.img of=/dev/sdb bs=512k
cgpt repair /dev/sdb
```

2

This is the point where you need to mark either boot partition 1 or 2 to have higher priority. The number with the higher priority will boot first. The example below will give priority 10 to the first partition (-i) and will thus boot successfully from a SD card.


```
cgpt add -i 1 -S 1 -T 5 -P 10 -l KERN-A /dev/sdb
cgpt add -i 2 -S 1 -T 5 -P 5 -l KERN-B /dev/sdb
```

To see your partition list and order, use the command **cgpt show**.

```
root@kali:~# cgpt show /dev/sdb
start size part contents
0 1 PMBR
1 1 Pri GPT header
2 32 Pri GPT table
8192 32768 1 Label: "KERN-A"
Type: ChromeOS kernel
UUID: 63AD6EC9-AD94-4B42-80E4-798BBE6BE46C
Attr: priority=10 tries=5 successful=1
40960 32768 2 Label: "KERN-B"
Type: ChromeOS kernel
UUID: 37CE46C9-0A7A-4994-80FC-9C0FFCB4FDC1
Attr: priority=5 tries=5 successful=1
73728 3832490 3 Label: "Linux filesystem"
Type: 0FC63DAF-8483-4772-8E79-3D69D8477DE4
UUID: E9E67EE1-C02E-481C-BA3F-18E721515DBB
125045391 32 Sec GPT table
125045423 1 Sec GPT header
root@kali:~#
```

Once this operation is complete, boot up your Samsung Chromebook with the SD/USB device plugged in. At the developer mode boot screen, hit CTRL+u to boot from from your USB storage device. Log in to Kali (root / toor) and startx.

Custom MK/SS808 Image

The following document describes our own method of creating a **custom Kali Linux MK/SS808 ARM image** and is targeted at developers. If you would like to install a pre-made Kali image, check out our [Install Kali on MK/SS808](#) article.

01. Create a Kali rootfs

Build yourself a [Kali rootfs](#) as described in our Kali documentation, using an **armhf** architecture. By the end of this process, you should have a populated rootfs directory in **~/arm-stuff/rootfs/kali-armhf**.

02. Create the Image File

Next, we create the physical image file which will hold our MK/SS808 rootfs and boot images.

```
apt-get install kpartx xz-utils sharutils
cd ~
mkdir -p arm-stuff
cd arm-stuff/
mkdir -p images
cd images
dd if=/dev/zero of=kali-custom-ss808.img bs=1MB count=7000
```

03. Partition and Mount the Image File

```
parted kali-custom-ss808.img --script -- mklabel msdos
parted kali-custom-ss808.img --script -- mkpart primary ext4 1 -1
```

```
loopdevice=`losetup -f --show kali-custom-ss808.img`
device=`kpartx -va $loopdevice| sed -E 's/.*(loop[0-9])p.*/\1/g' | head -1`
device="/dev/mapper/${device}"
rootp=${device}p1
```

```
mkfs.ext4 $rootp
mkdir -p root
mount $rootp root
```

04. Copy and Modify the Kali rootfs

```
rsync -HPavz /root/arm-stuff/rootfs/kali-armhf-xfce4/ root
echo nameserver 8.8.8.8 > root/etc/resolv.conf
```

05. Compile the rk3066 Kernel and Modules

If you're not using ARM hardware as the development environment, you will need to set up an [ARM cross-compilation environment](#) to build an ARM kernel and modules. Once that's done, proceed with the following steps.

```
apt-get install xz-utils
cd ~/arm-stuff
mkdir -p kernel
cd kernel

git clone git://github.com/aloksinha2001/picuntu-3.0.8-alok.git rk3066-kernel
cd rk3066-kernel
sed -i "/vpu_service/d" arch/arm/plat-rk/Makefile
```

```
export ARCH=arm
export CROSS_COMPILE=~/.arm-stuff/kernel/toolchains/arm-eabi-linaro-4.6.2/bin/arm-eabi-
```

```
# A basic configuration for the UG802 and MK802 III
# make rk30_hotdog_ti_defconfig
# A basic configuration for the MK808
```

```
make rk30_hotdog_defconfig

# configure your kernel !
make menuconfig
# Configure the kernel as per http://www.armvtech.com/armvtechforum/viewtopic.php?f=66&t=835
mkdir ../initramfs/
wget http://208.88.127.99/initramfs.cpio -O ../initramfs/initramfs.cpio

mkdir -p ../patches
wget http://patches.aircrack-ng.org/mac80211.compat08082009.wl_frag+ack_v1.patch -O
../patches/mac80211.patch
wget http://patches.aircrack-ng.org/channel-negative-one-maxim.patch- O ../patches/negative.patch
patch -p1 < ../patches/mac80211.patch
patch -p1 < ../patches/negative.patch

./make_kernel_ruikemei.sh
```

```
make modules -j$(cat /proc/cpuinfo|grep processor|wc -l)
make modules_install INSTALL_MOD_PATH=~/.arm-stuff/images/root
git clone git://git.kernel.org/pub/scm/linux/kernel/git/dwmw2/linux-firmware.git firmware-git
mkdir -p ~/.arm-stuff/images/root/lib/firmware
cp -rf firmware-git/* ~/.arm-stuff/images/root/lib/firmware/
rm -rf firmware-git
```

```
umount $rootp
kpartx -dv $loopdevice
losetup -d $loopdevice
```

07. dd the Image to a USB device

Use the **dd** utility to image this file to your SD card. In our example, we assume the storage device is located at `/dev/sdb`. **Change this as needed.**

```
dd if=kali-custom-ss808.img of=/dev/sdb bs=512k
```

Once the dd operation is complete, unmount and eject the SD card and boot your MK/SS808 into Kali Linux

Custom ODROID X2 U2 Image

The following document describes our own method of creating a **custom Kali Linux ODROID image** and is targeted at developers. If you would like to install a pre-made Kali ODROID image, check our [Install Kali on ODROID](#) article.

01. Create a Kali rootfs

Start by building a [Kali rootfs](#) as described in our Kali documentation using an **armhf** architecture. By the end of this process, you should have a populated rootfs directory in **~/arm-stuff/rootfs/kali-armhf**.

02. Create the Image File

Next, we create the physical image file which will hold our ODROID rootfs and boot images.

```
apt-get install kpartx xz-utils uboot-mkimage
cd ~
mkdir -p arm-stuff
cd arm-stuff/
mkdir -p images
cd images
dd if=/dev/zero of=kali-custom-odroid.img bs=1MB count=7000
```

03. Partition and Mount the Image File

```
parted kali-custom-odroid.img --script -- mklabel msdos
parted kali-custom-odroid.img --script -- mkpart primary fat32 4096s 266239s
parted kali-custom-odroid.img --script -- mkpart primary ext4 266240s 100%

loopdevice=`losetup -f --show kali-custom-odroid.img`
device=`kpartx -va $loopdevice| sed -E 's/.*(loop[0-9])p.*\1/g' | head -1`
device="/dev/mapper/${device}"
bootp=${device}p1
rootp=${device}p2
mkfs.vfat $bootp
mkfs.ext4 -L kaliroot $rootp
```

```
mkdir -p boot root
mount $bootp boot
mount $rootp root
```

04. Copy and Modify the Kali rootfs

Copy over the Kali rootfs you bootstrapped earlier using **rsync** to the mounted image.

```
cd ~/arm-stuff/images/
rsync -HPavz ~/arm-stuff/rootfs/kali-armhf/ root
echo nameserver 8.8.8.8 > root/etc/resolv.conf
```

Edit the `~/arm-stuff/images/root/etc/inittab` file and locate the “Example how to put a getty on a serial line”.

```
nano root/etc/inittab
```

Add the following line to the end of that section.

```
T1:12345:respawn:/sbin/agetty 115200 ttySAC1 vt100
```

If you want the serial console to autologin as root, use the following line instead:

```
T1:12345:respawn:/bin/login -f root ttySAC1 /dev/ttySAC1 >&1
```

Now, make sure there is a `ttySAC1` entry in the `~/arm-stuff/images/root/etc/udev/links.conf` file.

```
nano root/etc/udev/links.conf
```

If an entry for `ttySAC1` doesn't already exist, add it to the file so it looks as follows:

```
M null c 1 3
M console c 5 1
M ttySAC1 c 5 1
```

Add `ttySAC` entries in the `~/arm-stuff/images/root/etc/udev/links.conf` file.

```
cat << EOF >> root/etc/securetty
ttySAC0
ttySAC1
ttySAC2
EOF
```

Place a basic `xorg.conf` file in the rootfs.


```
cat << EOF > root/etc/X11/xorg.conf
# X.Org X server configuration file for xfree86-video-mali

Section "Device"
Identifier "Mali-Fbdev"
# Driver "mali"
Option "fbdev" "/dev/fb1"
Option "DRI2" "true"
Option "DRI2_PAGE_FLIP" "true"
Option "DRI2_WAIT_VSYNC" "true"
Option "UMP_CACHED" "true"
Option "UMP_LOCK" "false"
EndSection

Section "Screen"
Identifier "Mali-Screen"
Device "Mali-Fbdev"
DefaultDepth 24
EndSection

Section "DRI"
Mode 0666
EndSection
EOF
```

Link **init** in the root, rootfs directory:

```
cd ~/arm-stuff/images/root
ln -s /sbin/init init
```

05. Compile the ODROID Kernel and Modules

If you're not using ARM hardware as the development environment, you will need to set up an [ARM cross-compilation environment](#) to build an ARM kernel and modules. Once that's done, proceed with the following instructions.

We next need to fetch the ODROID kernel sources and place them in our development tree structure:

```
cd ~/arm-stuff
mkdir -p kernel
cd kernel
git clone --depth 1 https://github.com/hardkernel/linux.git -b odroid-3.8.y odroid
cd odroid
touch .scmversion
```

Configure, then cross-compile the ODROID kernel.

```
export ARCH=arm
export CROSS_COMPILE=~/.arm-stuff/kernel/toolchains/arm-eabi-linaro-4.6.2/bin/arm-eabi-

# for ODROID-X2
make odroidx2_defconfig
# for ODROID-U2
make odroidu2_defconfig
# configure your kernel !
make menuconfig
# and enable
CONFIG_HAVE_KERNEL_LZMA=y
CONFIG_RD_LZMA=y

# If cross compiling, run this once
sed -i 's/if defined(__linux__)/if defined(__linux__) || defined(__KERNEL__) /g' include/uapi/drm/drm.h

make -j $(cat /proc/cpuinfo|grep processor|wc -l)
make modules_install INSTALL_MOD_PATH=~/.arm-stuff/images/root/
```

Chroot into the rootfs and create an [initrd](#). Make sure to use the correct kernel version/extraversion for the **mkinitramfs** command. In our case, it was "3.8.13".

```
LANG=C chroot ~/arm-stuff/images/root/  
apt-get install initramfs-tools uboot-mkimage  
cd /  
# Change the example "3.8.13" to your current odroid kernel revision  
mkinitramfs -c lzma -o ./initramfs 3.8.13  
mkimage -A arm -O linux -T ramdisk -C none -a 0 -e 0 -n initramfs -d ./initramfs ./ulnitrd  
rm initramfs  
exit
```

06. Prepare the Boot Partition

Copy the kernel and generated initrd file to the mounted boot partition as shown below.

```
mv ~/arm-stuff/images/root/ulnitrd ~/arm-stuff/images/boot/  
cp arch/arm/boot/zImage ~/arm-stuff/images/boot/
```

Dump a **boot.txt** file, which contains required boot parameters for the ODROID in the boot partition.

```
cat << EOF > ~/arm-stuff/images/boot/boot.txt  
setenv initrd_high "0xffffffff"  
setenv fdt_high "0xffffffff"  
setenv bootcmd "fatload mmc 0:1 0x40008000 zImage; fatload mmc 0:1 0x42000000 ulnitrd; bootm  
0x40008000 0x42000000"  
setenv bootargs "console=tty1 console=ttySAC1,115200n8 root=LABEL=kaliroot rootwait ro  
mem=2047M"
```

```
boot
EOF
```

Generate a **boot.scr** file, which is required to boot the ODROID.

```
mkimage -A arm -T script -C none -n "Boot.scr for ODROID" -d ~/arm-stuff/images/boot/boot.txt
~/arm-stuff/images/boot/boot.scr
```

Unmount the root and boot partitions, then unmount the loop device.

```
cd ~/arm-stuff/images/
umount $bootp
umount $rootp
kpartx -dv $loopdevice

wget http://www.mdrjr.net/odroid/mirror/old-releases/BSPs/Alpha4/unpacked/boot.tar.gz
tar xzpf boot.tar.gz
cd boot
sh sd_fusing.sh $loopdevice
cd ..
losetup -d $loopdevice
```

Now, image the file onto your USB storage device. Our device is **/dev/sdb**. Change this as needed.

```
dd if=kali-custom-odroid.img of=/dev/sdb bs=1M
```

Once this operation is complete, connect your UART serial cable to the ODROID and boot it up with the microSD/SD card plugged in. Through the serial console, you will be able to log in to Kali (root / toor) and startx.

If everything works and you want the ODROID to start on boot, make sure to use the “autologin” line in the inittab given above and add the following to your bash_profile:

```
# If you don't have a .bash_profile, copy it from /etc/skel/.profile first
cat << EOF >> ~/.bash_profile
if [ -z "$DISPLAY" ] && [ $(tty) = /dev/ttySAC1 ]; then
startx
fi
EOF
```

08. Install Mali Graphic Drivers (Optional)

These steps are experimental and not fully tested yet. They should be preformed inside the Kali rootfs.

```
# http://malideveloper.arm.com/develop-for-mali/drivers/open-source-mali-gpus-linux-exadri2-and-
x11-display-drivers/
apt-get install build-essential autoconf automake make libtool xorg xorg-dev xutils-dev libdrm-dev
wget
http://malideveloper.arm.com/downloads/drivers/DX910/r3p2-01rel0/DX910-SW-99003-r3p2-01rel0.tgz
wget
http://malideveloper.arm.com/downloads/drivers/DX910/r3p2-01rel0/DX910-SW-99006-r3p2-01rel0.tgz
wget --no-check-certificate https://dl.dropbox.com/u/65312725/mali_opengl_hf_lib.tgz

tar -xzvf mali_opengl_hf_lib.tgz
cp mali_opengl_hf_lib/* /usr/lib/

tar -xzvf DX910-SW-99003-r3p2-01rel0.tgz
tar -xzvf DX910-SW-99006-r3p2-01rel0.tgz
cd DX910-SW-99003-r3p2-01rel0/x11/xf86-video-mali-0.0.1/
./autogen.sh
```

```
chmod +x configure
```

```
CFLAGS="-O3 -Wall -W -Wextra -I/usr/include/libdrm  
-IDX910-SW-99006-r3p2-01rel0/driver/src/ump/include" LDFLAGS="-L/usr/lib -IMali -IUMP -lpthread"  
./configure --prefix=/usr --x-includes=/usr/include --x-libraries=/usr/lib  
cp -rf ../../DX910-SW-99006-r3p2-01rel0/driver/src/ump/include/ump src/  
mkdir -p umplock  
cd umplock  
wget  
http://service.i-onik.de/a10_source_1.5/lichee/linux-3.0/modules/mali/DX910-SW-99002  
-r3p0-04rel0/driver/src/devicedrv/umplock/umplock_ioctl.h  
cd ..  
  
make  
make install
```

ARM Cross-Compilation

The following guide will demonstrate how to set up an ARM cross-compilation environment in Kali Linux. This guide is the starting point for many of our contributed “Custom ARM Images” articles.

Setting Up Your Development Box

Compiling kernels and generating images usually comes at the cost of disk space. Make sure you have at least 50 GB of disk space available on your Kali development machine as well as ample RAM and CPU juice.

Install Dependencies

Start off by installing the required dependencies for ARM cross-compilation.

```
apt-get install git-core gnupg flex bison gperf libbsd0-dev build-essential \  
zip curl libncurses5-dev zlib1g-dev libncurses5-dev gcc-multilib g++-multilib
```

If you are running a 64 bit Kali Linux system, add i386 architecture support to your development environment as follows.

```
dpkg --add-architecture i386  
apt-get update  
apt-get install ia32-libs
```

Download Linaro Toolchain

Download the Linaro cross-compiler from our Git repository.

```
cd ~  
mkdir -p arm-stuff/kernel/toolchains
```

```
cd arm-stuff/kernel/toolchains
git clone git://github.com/offensive-security/arm-eabi-linaro-4.6.2.git
```

Set Environment Variables

To use the Linaro cross-compiler, you will need to set the following environment variables in your session.

```
export ARCH=arm
export CROSS_COMPILE=~/.arm-stuff/kernel/toolchains/arm-eabi-linaro-4.6.2/bin/arm-eabi-
```

Now your ARM cross-compilation environment is complete and you can proceed with building your own ARM kernels.

Rebuilding a Package from Source

On occasion, we need to rebuild a Kali package from source. Fortunately, this is as simple as apt-getting the package sources, modifying them to your needs, and then rebuilding them using Debian tools. In this example, we will recompile the [libfreefare](#) package in order to add some extra hardcoded Mifare access keys into the mifare-format tool.

Downloading the Package Source

```
# Get the source package
apt-get source libfreefare
cd libfreefare-0.3.4~svn1469/
```

Edit the Package Source Code

Make the changes needed to the source code of the package. In our case, we modify an example file, mifare-classic-format.c.

```
nano examples/mifare-classic-format.c
```

Check for Build Dependencies

Check for any build dependencies the package may have. These need to be installed before you can build the package.

```
dpkg-checkbuilddeps
```

The output should be similar to the following, depending on what packages you already have installed. **fdpkg-**

dpkg-checkbuilddeps returns no output, that means you do not have any missing dependencies and can proceed with the build.

```
dpkg-checkbuilddeps: Unmet build dependencies: dh-autoreconf libnfc-dev
```

Install Build Dependencies

Install any build dependencies if needed, as shown in the output of **dpkg-checkbuilddeps**:

```
apt-get install dh-autoreconf libnfc-dev
```

Build the Modified Package

With all of the dependencies installed, it is a simple matter of invoking **dpkg-buildpackage** to build your new version.

```
dpkg-buildpackage
```

Install the New Package

If all went well, you should be able to install your newly-created package.

```
dpkg -i ../libfreefare*.deb
```

Recompiling the Kali Linux Kernel

On occasion, you might want to add certain drivers, patches, or kernel features that are not included in the stock Kali Linux kernel. The following guide will describe how the Kali Linux kernel can be quickly modified and recompiled for your needs. Please note that global wireless injection patches are already present by default in the Kali Linux kernel.

Install Build Dependencies

Start by installing all the build dependencies required for recompiling your kernel.

```
apt-get install kernel-package ncurses-dev fakeroot bzip2
```

Download Kali Linux Kernel Source Code

Download and extract the Kali Linux kernel source.

```
apt-get install linux-source  
cd /usr/src/  
tar jxpf linux-source-3.7.tar.bz2  
cd linux-source-3.7/
```

Configure Your Kernel

Copy over the default Kali kernel `.config` file and then modify it to your needs. This is the stage where you would apply various patches, etc. In this example, we are re-compiling a 64 bit kernel.

```
cp /boot/config-3.7-trunk-amd64 .config  
make menuconfig
```

Build the Kernel

Compile your modified kernel image. Depending on your hardware, this could take a while.

```
export CONCURRENCY_LEVEL=$(cat /proc/cpuinfo|grep processor|wc -l)
make-kpkg clean
fakeroot make-kpkg kernel_image
```

Install the Kernel

Once the kernel has successfully compiled, go ahead and install the new kernel and reboot. Please note that kernel version numbers may change - in our example, it was 3.7.2. Depending on the current kernel version, you might need to adjust it accordingly.

```
dpkg -i ../linux-image-3.7.2_3.7.2-10.00.Custom_amd64.deb
update-initramfs -c -k 3.7.2
update-grub2
reboot
```

Once rebooted, your new kernel should be running. If things go wrong and your kernel does not boot, you can still boot the original stock Kali kernel and fix your issues.

08. Troubleshooting Kali Linux

Submitting Bugs for Kali Linux

Introduction

This document will guide a reporter on how best to present a bug report so it gets addressed as quickly as possible. The goal of a bug report is to enable the Kali Linux developers to reproduce the issue and see the failure. If the Kali developers can make it fail, they will work to gather extra information until the root cause is known. If the Kali developers are unable to reproduce the failure, they will require additional information until they experience the same results as the submitter. Please note, submissions are best read by our team in **english**.

Kali Linux was born out of a labor of love to give back to the community. It is our drive to make things better for everyone which keeps this project evolving. The developers who provide support to you, are volunteers doing so out of altruism. Please keep this in mind when making your comments.

Here are a few pointers that will lead to success in getting issues resolved:

- You are reporting the bug because you want it fixed, supply all the information you can.
- Make it very clear in your submission what are facts and what are hypotheses.
- Keep the bug report objective, just the facts needed for proper research.
- Do not quote Wikipedia and other non-primary resource as fact in your submission.
- One report, per person, per hardware combination, per bug.
- Do not stack multiple issues into a single report, submit additional reports as needed.
- Do not post comments that are unhelpful such as “Me too!” or “+1”
- Do not complain about how long it takes to fix a bug.

How to report bugs

The Kali Linux bug tracker can be found at <http://bugs.kali.org>. This document will guide you through account creation, creating a system profile, and how to submit a detailed report for submission to the bug tracker.

Create a Kali Linux Bug Tracker Account

If you have not already created an account, you will need to complete this first. Creating the account will allow you to submit reports and comment on existing ones.

On the bug tracker website, click ‘Signup for new account’ to begin the process.

KALI LINUX BUG TRACKER

Anonymous | [Login](#) | [Signup for a new account](#)

2013-03-20 05:25 EDT

[Main](#) | [My View](#) | [View Issues](#) | [Change Log](#) | [Roadmap](#) | [Repositories](#)

Unassigned [^] (1 - 10 / 47)

0000147 —	syslinux.cfg contains a few mistakes [All Projects] General Bug - 2013-03-19 21:38
0000146 ^	The debian openssl has a --no-sslv2 patch [All Projects] Kali Package Bug - 2013-03-19 15:42
0000143 —	Automated HTTP Enumeration Tool [All Projects] New Tool Requests - 2013-03-19 14:40
0000142 —	Unhide Forensic Tool, Find hidden processes and ports [All Projects] New Tool Requests - 2013-03-19 14:39
0000140 —	Inguma [All Projects] New Tool Requests - 2013-03-19 14:37
0000139 —	Junkie [All Projects] New Tool Requests - 2013-03-19 14:36
0000138 —	sqlmap [All Projects] Tool Upgrade - 2013-03-19 14:08
0000135 —	android-sdk issue [All Projects] General Bug - 2013-03-19 13:01
0000130	Need to upgrade python-usb from 0.8 to 1.0 for ubertooth software

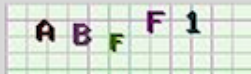
Resolved [^] (1 - 5 / 5)

0000122 —	msfpro console fails to launch [All Projects] General Bug - 2013-03-19 13:01
0000076 —	b43 wireless driver firmware r [All Projects] Kali Package Bug - 2013-03-19 12:00
0000102 —	The Social-Engineer Toolkit (SE [All Projects] Tool Upgrade - 2013-03-19 11:00
0000100 —	Social Engineering Tool cannot [All Projects] General Bug - 2013-03-19 10:00
0000063 ^ U	No Keyboard or Mouse after M [All Projects] General Bug - 2013-03-19 09:00

You will need to provide a username, e-mail address and enter the information from the captcha challenge. Click the signup button to proceed.

KALI LINUX BUG TRACKER

Signup

Username:	<input type="text" value="NewBugSubmitter"/>
E-mail:	<input type="text" value="nbs@email.com"/>
Enter the code as it is shown in the box on the right.:	<input type="text" value="ABFF1"/> 

On completion of this form and verification of your answers, you will be sent a confirmation e-mail to the e-mail address you specified. Using the confirmation e-mail, you will be able to activate your account. If you fail to activate your account within seven days, it will be purged. You must specify a valid e-mail address in order to receive the account confirmation e-mail.

[[Login](#)] [[Lost your password?](#)]

If successful, the next page will notify you that the account registration has been processed. You will need to respond to the confirmation email in order to have your account officially activated. Click 'Proceed' to continue to the Bug Tracker Login page.

KALI LINUX BUG TRACKER

Account registration processed.

Congratulations. You have registered successfully. You are now being sent a confirmation e-mail to verify your e-mail address. Visiting the link sent to you in this e-mail will activate your account.

You will have seven days to complete the account confirmation process; if you fail to complete account confirmation within seven days, this newly-registered account may be purged.

[[Proceed](#)]

Create a Profile in Kali Linux Bug Tracker

Although not required, it is recommended to create a unique profile as part of your bug tracker account. You can create a custom profile for each system, or select from the default profiles provided. These profiles are shortcuts to define the values for your Platform, OS and version information submitted as part of your bug report.

To create or edit a custom profile, select My Account from the main page and the select Profiles. Add the specific information and description for your system and click the Add Profile button when done.

Add Profile		[My Account]	[Preferences]	[Manage Columns]	[Profiles]
*Platform	<input type="text" value="Intel x64"/>				
*Operating System	<input type="text" value="Kali"/>				
*OS Version	<input type="text" value="1.0.1"/>				
Additional Description	<pre>Linux kali 3.7-trunk-amd64 #1 SMP Debian 3.7.2-0+kali6 x86_64 GNU/Linux -This system is a VMWare guest system -VMWare Fusion Professional Version 5.0.3 (1040386) -2 processor cores (2.6GHz Intel Core i7) -4096MB RAM</pre>				
*required		<input type="button" value="Add Profile"/>			

Edit or Delete Profiles	
<input checked="" type="radio"/> Edit Profile <input type="radio"/> Make Default <input type="radio"/> Delete Profile	
Select Profile	<input type="text"/>
<input type="button" value="Submit"/>	

Once the profile is added, it will appear in the 'Select Profile' dropdown list when you create a new Report Issue. You can create as many different profiles as you require, just ensure you select the appropriate one when submitting your bug report.

Ensure you are not duplicating a previous request

Before starting your report, search the site for keywords relating to your issue. If there is already an existing bug not related to hardware, please do not duplicate the request or add notes that are unnecessary. (i.e. "Me Too" or "+1") If it has been reported, you can view the status of the issue by clicking the ID link.

If you believe the issue to be hardware related, please submit a new report with your specific information, even if it appears similar. There is a strong chance that your hardware does not exactly match that of another report issuer. Do not assume because you have the same desktop or laptop model that your issue is not unique.

Creating the report

To begin your report, log into your account and click the “Report Issue” link on the landing page. You will need to fill out as much information as you possibly can. If it helps, review the pointers at the beginning of this document to make sure you are conforming to expectations.

The following fields are mandatory within the report:

- Category
- Summary
- Description

Even though the other fields are not mandatory, we recommend you try to include as much information as possible within each option paying special attention to the following:

- Reproducibility
- Select Profile
- Steps to Reproduce
- Additional Information
- Upload File (error logs, screenshot)

Decide the proper Category

There are currently four (4) categories available in the Kali bug tracker. Before you begin your request ensure it is properly designated for one of the following:

- General Bug
- Kali Package Bug
- New Tool Requests
- Tool Upgrade

Do not request support within the bug tracker. Kali Linux offers several options for support including <http://docs.kali.org> , <https://forums.kali.org> and the IRC chat room on freenode (#kali-linux)

Providing a Descriptive Summary

The summary field is essentially the ‘name’ of the report, it will be the first thing the Kali developers and other visitors see. Provide a short, yet descriptive summary that can describe the issue or request.

Good: Chromium Package installed from Repo will not run as root user

Bad: Chromium doesn't work

The summary does not need to include everything, but it should convey your reason for submitting the report.

Using dpkg to find the package and version for the report

You can find which package is installed using a combination of dpkg flags. It is important to include relevant information from the output of these commands in your report. The output can also be placed in a text file and uploaded. (Discussed later within this document.)

- search
- list
- status

Sample Output

```
root@kali:~# which chromium
/usr/bin/chromium
root@kali:~# type chromium
chromium is /usr/bin/chromium
root@kali:~# dpkg --search /usr/bin/chromium
chromium: /usr/bin/chromium
root@kali:~# dpkg --list chromium
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Conf-files/Unpacked/half-conf/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
||/ Name          Version      Architecture Description
+++=====
=====
ii chromium      24.0.1312.68 amd64      Google open source chromium web
root@kali:~# dpkg --status chromium
Package: chromium
Status: install ok installed
Priority: optional
Section: web
Installed-Size: 98439
Maintainer: Debian Chromium Maintainers <pkg-chromium-maint@lists.aliases.debian.org> Architecture:
amd64
Source: chromium-browser
```

```
Version: 24.0.1312.68-1
...Output Truncated...
```

Building the Description scenario

This is your opportunity to provide a well thought out description of what you are reporting. This is your chance to shine and provide as much details and facts as possible.

Please ensure you include the following where applicable:

- Exact and complete text of any error messages (screen output or log files)
- Exactly what you typed or actions you took to produce the issue
- A suggested fix, or patch if you are able to produce one
- The version of the package and any information relating to dependent packages
- The kernel version, shared C library, and any other details that seem appropriate
- `uname -a`
- `dpkg -s libc6 | grep ^Version`
- If applicable, software version - (i.e. `python -V`)
- Details of your hardware
- If you are reporting an issue with a device driver, please list all hardware in your system
- For a complete report on your system install `lshw` from the repos.
- Add any other details that seems relevant
- Do not worry about the report being “too long” as long as the information is relevant, it’s important to include.

Example

Package: Chromium

Architecture: amd64

Maintainer: Debian Chromium Maintainers

Source: chromium-browser

Version: 24.0.1312.68-1

I installed the chromium web browser from the Kali Linux repos, using the command ‘`apt-get install chromium`’. I launched the program from the Kali menu by selecting Applications/Internet/Chromium Web Browser. Chromium

did not launch as expected, instead it provided an error pop-up window.

The error message stated, "Chromium cannot be run as root. Please start Chromium as a normal user. To run as root, you must specify an alternate `-user-data-dir` for storage of profile information".

I clicked the Close button to close the pop up window.

```
uname -a output: Linux kali 3.7-trunk-amd64 #1 SMP Debian 3.7.2+kali6 x86_64 GNU/Linux
```

C Library Version: 2.13-38

The Importance of Reproducibility

The Kali Linux bug tracker allows you to provide the frequency of the issue being reported. If you are submitting a request for a new tool or an upgrade to an existing tool, simply select N/A from the drop down options. If submitting a bug, please provide the appropriate response.

Continuing the example above, by design Chromium will not launch as root, you would select 'always' from the dropdown menu.

It is extremely important you provide an accurate response, if the Kali developers attempt to reproduce the issue they need to know the frequency. If the issue happens occasionally, but you have marked always, the issue may be closed prematurely as the developer testing may not experience the issue.

Selecting the Proper Profile

As discussed above, using a custom defined profile is best for each issues reported. If custom profiles are not created, select the appropriate profile from the dropdown menu. At the time of this guide the following options are available.

- armel Kali 1.0
- armhf Kali 1.0
- x64 Kali 1.0
- x86 Kali 1.0

Providing steps to reproduce the Issue

Although this may seem redundant when compared with the description section, this section should only include the steps taken to reproduce the issue. Some steps may seem remedial, but it is important to ensure you document as well as you can. The missing step may be the one needed to reproduce the issue.

Example:

1. Opened a terminal window by selecting Applications/Accessories/Terminal
2. Typed 'apt-get install chromium' in the terminal and hit enter to run the command
3. Attempted to run Chromium web browser by selecting Applications/Internet/Chromium Web Browser

Providing Additional Information

In this section you can provide any additional information relevant to the issue. If you have a fix for the issue, please provide it in this portion. Again, it is important to stick to the facts and document the steps properly so the developers can reproduce.

Example:

There is a simple fix that is well documented on several forums. I tried it and it fixed the issue for me.

- Using a text editor open /etc/chromium/default
- Add -user-data-dir flag
- i.e. CHROMIUM_FLAGS="-user-data-dir"

Can this be patched within the repo version of Chromium so adding this flag is not required for future releases?

Uploading relevant files

Sometimes it is important to provide information to the development team that can't easily be provided. This section of the report allows you to add screenshots and log files. Be mindful the size limitation in place.

You can add a file by clicking the "Choose File" button. This will open the file manager for your system and allow you to select the file. Once you have selected the file click the "Open" button to return to your report and click the "Upload File" button.

Submitting the Report

If you've come this far, you are ready to submit the report. All that is left to do is click the "Submit Report" button. Your report will be submitted and assigned a tracking ID. The report will show up on your "My View" page under "Reported by Me." This will allow you to track the issue to resolution.

Summary

The purpose for a bug report is to help the developers see the failure with their own eyes. Since they cannot be

with you to experience the failure you must provide detailed instructions so they can make it fail themselves.

Describe everything in detail, stating the steps taken, what you saw, what you did as well as the expected outcome.

Attempt to find an issue or fix through research. If you are able to provide a solution to fix the issue for your system, provide the developers with the same level of detail for bug reporting. It is important that the developers know exactly what you did, so they can successfully repeat the process. This should not stop you from filing a full explanation of the symptom that caused the unexpected behavior.

Write accurately, be clear, precise and concise to ensure the developers cannot misinterpret what you are trying to convey.

No developer will be deliberately coy, be prepared to provide additional information, the developers will not ask if they don't need the information.

Please be patient with your request, the developers want to fix your issue as much as you do. We love what we do and are proud to continue making Kali the most advanced penetration testing distribution ever.

This articles has been composed from various resources listed below, and modified to suit our needs:

<http://www.chiark.greenend.org.uk/~sgtatham/bugs.html> - Fetched March 20,2013

<https://help.ubuntu.com/community/ReportingBugs> - Fetched March 20,2013

<http://www.debian.org/Bugs/Reporting> - Fetched March 20,2013

Troubleshooting Wireless Drivers

Troubleshooting wireless driver issues in Linux can be a frustrating experience if you don't know what to look for. This article is meant to be used as a general guideline to better help you find the information you need to solve your wireless issues.

Carefully read carefully ANY error message as they will VERY OFTEN tell you what's wrong and how to fix it. If not, then use your Google-Fu.

1. No Interface

- Stupid question: Is it a wireless card? (We've seen that several times)
- Is the device plugged in?
- Does it show up on **lsusb** or **lspci** (with the exception of phones)? You might want to update pci ids and usb ids
- Does **dmesg** contain any information about the driver loading and/or failing
- Is Kali a VM? Then, unless your card is USB, it will not be useable (VMWare/VirtualBox/QEMU will virtualize EVERY PCI device). Is it attached to the VM?
- If there is nothing in **dmesg** and it's not in a VM, then you might want to try the latest *compat-wireless* (and sometimes, you'll need firmware) -> check on Linux-Wireless drivers

2. Interface But Can't Do Anything

- Read error messages
- If there are no error messages, then run **dmesg | tail** and it will most likely tell you what's wrong
- Firmware might be missing
- Check rfkill and any hardware switches and BIOS options

3. No Monitor Mode

- STA drivers (Ralink, Broadcom) and every other manufacturer's provided driver doesn't support monitor mode
- ndiswrapper doesn't support monitor mode AND NEVER WILL.
- Airodump-ng/Wireshark don't show any packets: check rfkill and any hardware switches and BIOS options

4. Injection

- Test with aireplay-ng -9 (Make sure the card is in monitor mode with airmon-ng)
- Airmon-ng doesn't display chipset information: It's not a big issue as it just didn't get that information from the card and doesn't change the abilities of your card
- No injection but monitor mode: Check rkill and any hardware switches and BIOS options
- Network managers sometimes interfere with Aircrack tools. run **airmon-ng check kill** to kill these processes.

Additional Links

- [Will my card work with Aircrack-ng?](#)
- [Compat-wireless](#)

09. Kali Community Support

Kali Linux IRC Channel

Kali Linux has an official IRC channel located on the [Freenode](#) network. The official IRC channel is **#kali-linux**. Please take a few moments to review the rules and guidelines below before joining the channel.

IRC Rules and Guidelines

We try to remain as informal as possible but there are some rules that we'd appreciate if you would follow! Broadly, if you're friendly, tolerant, and reasonable, you'll probably get a long way without any specific knowledge of the rules - but for the avoidance of doubt, here they are.

How to Treat Other Users

In order to make the channel a pleasant place for all of our users, we expect all to remain as friendly and tolerant as possible. We request that you refrain from profanity and that you show respect to channel members and visitors. If you find that you're becoming frustrated with the channel or other users, we encourage you to take a moment to do something else. Try to ensure you don't make people feel like you're just taking advantage of them - help others out while you're waiting for a reply to your questions, and say thanks!

How to Argue

As mentioned above, we'd appreciate it if you'd strive to be friendly and tolerant. We also encourage debates and in-depth discussions about topical subjects. If you choose to participate in one, we expect you to remain as reasonable as possible and employ the skills of logic and critical thinking. These skills will serve you well in discussion, enable you to communicate more efficiently, and spot when others are being less than forthcoming with the truth!

Staying on Topic

We maintain no strict policy regarding off-topic chat in the channel however, the discussion of Kali Linux projects is the primary focus of the channel, so you may be asked to take discussions elsewhere, particularly if there are venues on freenode better suited to them (such as ##politics), if there are other conversations going on, or if they're repetitive or otherwise seen by the channel staff as being detrimental to the good atmosphere of the channel.

Certain things are seen as being specifically off-topic. These topics include:

Support or encouragement of illegal activity – it should go without saying, but we don't exist to help you break the law or do things you shouldn't be doing. Such queries are off-topic for the channel, for freenode as a whole, and may well get you removed from the channel and/or network. Please don't ask. Laws vary from country to country and channel OPs may determine whether a specific discussion is appropriate for the channel or not. Warez/Cracks/Pirated Software – these too are offtopic for the channel and network so please don't ask.

Political/Religious Matters

Many people have very strong political/religious beliefs and we respect that. We also recognize that these are volatile topics, which have nothing to do with Kali Linux and are best discussed elsewhere.

Asking for Help

If you're asking for help, first off, thanks! – questions and the resulting discussion of the answer(s) in a collaborative environment are what make IRC great and by helping to add to the atmosphere, you benefit all of us. We often find that we learn a lot even from questions we already think we know the answers to – about people, alternative approaches, and cool new resources and tools. However, if you are intending to ask a question, we'd appreciate it if you'd follow a couple useful guidelines to help you, and us, make the best use of our time: Do your research first – it's very frustrating when people ask a question that can virtually be answered by punching the keywords into a Google search! We also have forums and a wiki that contain answers to many questions we see daily so it's to everyone's benefit if these assets are used before asking in IRC.

Give Us the Whole Picture

If you're asked for more information, please provide it accurately. The correct answer will depend on it. Looking at this from another angle: the more we learn about your problem, the more this independently benefits us too – a large part of the development of new releases is derived from helping others with issues discovered with specific setups; even if you're asking us questions, you can help teach us something too! If you find the answer somewhere else, tell us – it isn't compulsory, but if you don't get an answer to your question in the channel but you find it elsewhere, consider letting us know! That way, we can help out the next person with a similar question. It also lets people know that you already have an answer you're happy with, or that if anyone's researching the question for you, they can stop. Wait for the answer – not everyone in the channel is online all the time – you may find you get an answer a few minutes, or even hours, later. Feel free to stick around and chat, or even answer other peoples questions – you'll find it helps pass the time and makes others likely to help you! Help us build a community of friendly security professionals and enthusiasts.

Spam, Flooding, and Various Other Forms of Disruptive Behaviour

Spam, flooding, disrespect or verbal attacks against other users, misleading links, intentionally wrong answers, and other forms of disruptive behaviour not otherwise specified are unwelcome. Disruptive behaviour includes

but is not restricted to the operation of unauthorised bots, public logging of the channel, and scripts such as those that publicly announce what track your MP3 player is playing or your away status. If you have more than 5 lines of text to paste, use [pastebin](#) for your data and then paste the URL into channel.

Dealing With the Channel Staff

From time to time, you may be asked to take conversations elsewhere, treat others reasonably, steer a conversation in a particular direction, or a variety of other things in order to preserve the ambiance and usefulness of the channel. If you're the target of such a request, please be as reasonable as you can and if you wish to take issue with it, do so in a private message with the channel staffer in question, rather than making noise in channel.

Discipline

Repeated breaking of the rules will cause channel staffers to mute (+q), ban (+b), kick, or otherwise remove you from the channel. This will particularly apply if you're seen to be willfully ignoring the rules after we've drawn your attention to them. Many forms of disruptive behaviour, such as flooding or trolling, may result in discipline without a warning. We try and avoid the use of force wherever possible and we'd appreciate it if you'd help us in pursuing this goal! If you're a bystander while a staffer is forced to use his or her powers for channel management, we'd appreciate your understanding and consideration in awaiting the end of the incident, and your assistance in keeping the situation as favorable as possible by not complaining, commentating, or gloating. This serves to make antisocial behavior such as flooding less attractive (the smaller the reaction, the less the return on the malfeasance), and so benefits you as well as us!

Official Kali Linux Mirrors

Using Official Repositories

Kali Linux provides three repositories, which are mirrored world-wide:

- [http.kali.org](http://kali.org) ([mirrorlist](#)): the main package repository;
- security.kali.org ([mirrorlist](#)): the security package repository;
- cdimage.kali.org ([mirrorlist](#)): the repository with ISO images.

When you use the above 3 hosts, you'll automatically be redirected to a mirror close to you, which is guaranteed to be up-to-date. If you prefer to manually select a mirror, click on the *mirrorlist* link near each hostname above and select a mirror that suits you.

Setting Up a Kali Linux Mirror

Requirements

To be an official Kali Linux mirror, you need a server with lots of disk space, good bandwidth, rsync, and an SSH server. As of 2013-03-14, the main package repository is about 160 GB and the ISO images repository is about 10 GB but you can expect those numbers to grow regularly. You are expected to make the files available over HTTP and RSYNC so you will need the corresponding services too. FTP access is optional.

Push Mirroring of Package Archives

The mirroring infrastructure uses SSH-based triggers to ping the mirrors when they need to be refreshed. This currently takes place 4 times a day.

If you don't have yet an account dedicated for the mirrors, create such an account (here we call it "archvsync"):

```
$ sudo adduser --disabled-password archvsync
Adding user 'archvsync' ...
[...]
Is the information correct? [Y/n]
```

Create the directories that will contain the mirrors and change their owner to the dedicated user that you just created:

```
$ sudo mkdir /srv/mirrors/kali{-security,-images}
$ sudo chown archvsync:archvsync /srv/mirrors/kali{-security,-images}
```

Next, configure the rsync daemon (enable it if needed) to export those directories:

```
$ sudo sed -i -e "s/RSYNC_ENABLE=false/RSYNC_ENABLE=true/" /etc/default/rsync
$ sudo vim /etc/rsyncd.conf
$ cat /etc/rsyncd.conf
uid = nobody
gid = nogroup
max connections = 25
socket options = SO_KEEPALIVE

[kali]
path = /srv/mirrors/kali
comment = The Kali Archive
read only = true

[kali-security]
path = /srv/mirrors/kali-security
comment = The Kali security archive
read only = true

[kali-images]
path = /srv/mirrors/kali-images
comment = The Kali ISO images
read only = true
$ sudo service rsync start
Starting rsync daemon: rsync.
```

This tutorial doesn't cover the configuration of the web server and the FTP server. Ideally, you should export the mirrors at <http://yourmirror.net/kali>, <http://yourmirror.net/kali-security> and <http://yourmirror.net/kali-images> (same for FTP). Now comes interesting part: the configuration of the dedicated user that will handle the SSH trigger and the actual mirroring. You should first unpack [ftpsync.tar.gz](http://archive.kali.org/ftpsync.tar.gz) in the user's account:

```
$ sudo su - archvsync
$ wget http://archive.kali.org/ftpsync.tar.gz
$ tar xzf ftpsync.tar.gz
```

Now we need to create two configurations files. We start from a template and we edit at least the *MIRRORNAME*, *TO*, *RSYNC_PATH*, and *RSYNC_HOST* parameters:

```
$ cp etc/ftpsync.conf.sample etc/ftpsync-kali.conf
$ cp etc/ftpsync.conf.sample etc/ftpsync-kali-security.conf
$ vim etc/ftpsync-kali.conf
$ grep -E '^[^#]' etc/ftpsync-kali.conf
MIRRORNAME=`hostname -f`
TO="/srv/mirrors/kali/"
RSYNC_PATH="kali"
RSYNC_HOST=archive.kali.org
$ vim etc/ftpsync-kali-security.conf
$ grep -E '^[^#]' etc/ftpsync-kali-security.conf
MIRRORNAME=`hostname -f`
TO="/srv/mirrors/kali-security/"
RSYNC_PATH="kali-security"
RSYNC_HOST=archive.kali.org
```

The last step is to setup the `.ssh/authorized_keys` file so that `archive.kali.org` can trigger your mirror:

```
$ mkdir -p .ssh
$ wget -O - -q http://archive.kali.org/pushmirror.pub >>.ssh/authorized_keys
```

If you have not unpacked the `ftpsync.tar.gz` in the home directory, then you must adjust accordingly the “`~/bin/ftpsync`” path, which is hard-coded in `.ssh/authorized_keys`. Now you must send an email to devel@kali.org with all the URLs of your mirrors so that you can be added in the main mirror list and to open up your `rsync` access on `archive.kali.org`. Please indicate clearly who should be contacted in case of problems (or if changes must be made/coordinated to the mirror setup). Instead of waiting for the first push from `archive.kali.org`, you should run an initial `rsync` with a mirror close to you, using the mirror list linked above to select one. Assuming that you picked `archive-4.kali.org`, here’s what you can run as your dedicated mirror user:

```
$ rsync -qaH archive-4.kali.org::kali /srv/mirrors/kali/ &
$ rsync -qaH archive-4.kali.org::kali-security /srv/mirrors/kali-security/ &
$ rsync -qaH archive-4.kali.org::kali-images /srv/mirrors/kali-images/ &
```

Manual Mirror of ISO Images

The ISO images repository does not use push mirroring so you must schedule a daily `rsync` run. We provide a `bin/mirror-kali-images` script, which is ready to use that you can add in the crontab of your dedicated user. You just have to configure `etc/mirror-kali-images.conf`.

```
$ sudo su - archvsync
$ cp etc/mirror-kali-images.conf.sample etc/mirror-kali-images.conf
$ vim etc/mirror-kali-images.conf
$ grep -E '^[^#]' etc/mirror-kali-images.conf
TO=/srv/mirrors/kali-images/
$ crontab -e
$ crontab -l
# m h dom mon dow command
```



```
39 3 * * * ~/bin/mirror-kali-images
```

Please adjust the precise time so that archive.kali.org doesn't get overloaded by too many mirrors at the same time.

Official Kali Linux Sites

[Kali Linux](#) has a number of sites available to serve our users. Listed below are the official Kali sites and the purpose each serves. Note that these sites are the **only** official Kali Linux sites and are the only authoritative sources of information available for the distribution.

The sites listed below are the **ONLY** official outlets for the Kali Linux Distribution.

Public Websites

- www.kali.org
- docs.kali.org
- forums.kali.org
- bugs.kali.org
- git.kali.org

The main [Kali Linux website](#) is our primary means of communicating about Kali Linux news, basic information, and updates about our project in general. It is here that you will find blog posts about new tools, features, and tricks about Kali Linux and this should be your one and only source to [download](#) the distribution.

This is where you are right now. Our documentation site contains a basic set of Kali Linux related documentation and tutorials. The changes that have been introduced with Kali are substantial and we have tried to cover a wide range of commonly asked questions. Sub-domains of docs.kali.org are also considered official (document translation servers).

Should you encounter an issue or situation that isn't covered in the [official Kali Linux documentation](#), there is a very high likelihood that there is a member of the [Kali Linux Forums](#) knows the answer. You will find that the Kali forum members are from all over the world, cover the entire range of skill levels, and are open and willing to help newcomers who are willing to learn.

Despite our best efforts at making Kali Linux perfect, unanticipated bugs and errors are inevitable. We are always open to improvement and can only effectively do so when issues or tools suggestions are reported to us. You are encouraged to submit bug reports at bugs.kali.org to help us make Kali Linux even better.

For our users that wish to closely monitor the development of Kali Linux or for people who want to know when they should run 'apt-get upgrade', our Git repository tree is available for perusal by the public.

Social Media

- [twitter](#)
- [facebook](#)

We don't tweet a lot but when we do, it's important. Information on releases and blog posts will be pushed to our twitter account, [@KaliLinux](#).

As with our Twitter account, we won't overwhelm you with information on our [Kali Facebook page](#) but when we do post, it will be worth it.

Kali Linux Bug Tracker

Kali Linux has an official [bug tracker](#) where our users can submit bugs and/or fixes to the developers and suggest new tools for inclusion in the distribution. Anyone can register for this site, but we ask that you please review the rules below to ensure bugs are submitted properly, with the correct information, and in the proper format.

- The Bug Tracker is NOT for support issues.
- Use a real email address so we can contact you if we need further clarification.
- Provide a descriptive subject.
- Provide as much detail as possible, including console output, architecture type, and exact versions.
- Tool requests must be accompanied by a URL and justification for adding the tool.
- Do not assign your bug to anyone. Developers will determine who is assigned the ticket.

Kali Linux Community Forums

Kali Linux has official community-driven forums located at forums.kali.org

We welcome everyone to the Kali Linux community and we have outlined a few simple rules below. Please take a few moments to review them before joining the forums.

Forum Rules

- By registering with our forums you agree to be bound by the following rules.
- We do not condone any illegal activity at all.
- Any advice/information offered in the forums is to be used for the legal informational/professional/educational purposes for which it is intended.
- New registrants posts will be moderated first, causing a slight delay in the post appearing - DO NOT report problems with your post not appearing instantly during your first 3 days of membership.
- Use sensible descriptive titles for your posts - not titles such as "Please Help!!" or "Need Assistance" or "what Am I Doing Wrong?" etc
- Do not cross-post - 1 post in the relevant area is enough!
- Search for related previous postings before creating a new thread. If you create a new thread asking something that has already been asked, don't be surprised if the thread gets deleted without notice.
- Do not post about breaking into networks that do not belong to you and for which you have no permissions.
- Any religious, political, or pornographic references will not be tolerated.
- Posts like - "Oooh! look!! I've cracked my neighbours wireless AP" or "How do I hack a network!?" are not needed here, thanks.
- Please don't bother with spam messages - they will be removed/moved/edited/deleted and you will be banned.
- Members signatures may NOT contain URL links, in any form.
- We will not tolerate abusive, sexist, racist, or any other derogatory remarks, or members acting like self-appointed moderators. The forum staff are here to help you. Please use their services. If ANY member has an issue with the content of ANY post within the forums, use the "REPORT THIS POST" button - This is the red triangle icon when using the default forum theme, or the asterisk icon when using the Blackfire Razor forum theme, found in the top right corner of each post.
- Breaking the forum rules may incur infractions ranging from loss of posting privileges to a temporary or permanent ban.
- These rules are subject to alteration and/or addition. It is your responsibility to be aware of any changes.

10. Kali Linux Policies

Kali's Relationship With Debian

Kali Linux 1.0 is a Debian derivative based on [Debian Wheezy](#). Therefore, most of the Kali packages are imported unmodified from the Debian repositories. In some cases, newer packages got imported from Unstable or Experimental, either because it improved the user experience, or because it was required to fix some bugs.

Forked Packages

Some packages obviously had to be forked in order to implement some of the Kali-specific features but Kali strives to keep that number of packages to a minimum by improving the upstream packages when possible (either by integrating the feature directly, or by adding the required hooks so that it's trivial to enable it without actually modifying the upstream packages).

Each package forked by Kali is maintained in a [Git repository](#) with a "debian" branch so that updating a forked package can be easily done with a simple `git merge debian` in the master branch.

New Packages

On top of this, Kali brings many new Debian packages, which are specific to the penetration testing and security auditing field. A large percentage of these packages are free according to [Debian's Free Software Guidelines](#) and Kali intends to contribute those to Debian and to maintain them directly within Debian.

As a consequence of this, Kali packages strive to be compliant to the [Debian Policy](#) and follow the best practices in use in Debian.

Kali Linux Open Source Policy

Kali Linux is a Linux distribution that aggregates thousands of free software packages in its `main` section. As a Debian derivative, all of the software therein complies with the [Debian Free Software Guidelines](#).

As an exception to the above, Kali Linux's `non-free` section contains several tools which are not open source, but allowed for redistribution by [Offensive Security](#) through default licensing or specific license agreements with said vendors. If you want to build a Kali derivative, you should review the license of each Kali-specific non-free package before including it in your distribution (non-free packages which are imported from Debian are safe to redistribute).

More importantly, all of the specific developments that Kali made for its infrastructure or to integrate the provided software have been put under the [GNU GPL](#).

If you want more information about the license of any given piece of software, you can either check `debian/copyright` in the source package or `/usr/share/doc/package/copyright` for a package that you have already installed.

Kali Linux Trademark Policy

[Kali Linux](#) and [Offensive Security](#) want to promote the widespread recognition of our trademarks among the Internet community however, we also need to ensure our trademarks solely identify our company and our products. At the heart of our trademark policy is **trust** - we want to avoid the public from being confused into believing they are dealing with Kali Linux and/or Offensive Security when, in fact, they are not. This is of particular importance with regards to the development and distribution of trusted penetration testing distribution such as Kali Linux.

This document identifies and the describes our trademarks and provides guidance as to their fair use. We are generally quite accommodating when it comes to fair and honest use of our trademarks so if you are so inclined, feel free to contact us for further guidance.

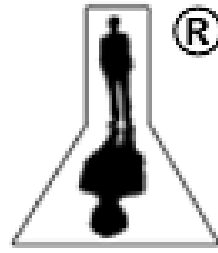
Some of our Trademarks

KALI LINUX™

OFFENSIVE®
SECURITY



TRY HARDER®



Use in Print, Web, Media and Public Display

It is important to maintain the look and spelling of the trademarks. Please do not modify the marks. Examples of modifying the marks include abbreviating names, adding logos to the marks, or combining the marks with other words. We recommend you use the trademarks in the exact form as we use them.

The Offensive Security trademarks are to designate the source of our products and services. We encourage others to use the marks so long as they are used to identify the products and services of Offensive Security. We do not want to confuse the public into believing that they are dealing with us, when in fact, they are not.

The first mention of an Offensive Security trademark should be accompanied by a symbol indicating whether the mark is a registered trademark “®” or an unregistered trademark “™”. Please refer to the above list for the appropriate symbol to use and if in doubt, use “™”.

The use of an Offensive Security trademark should be set apart from surrounding text, either by capitalizing it or by italicizing, bolding or underlining it. The Offensive Security trademarks are to designate the source of our products and services.

When using an Offensive Security trademark in written materials, you should provide a statement indicating that the [trademark] is a trademark of Offensive Security. For example:

“KALI LINUX™ is a trademark of Offensive Security.” This statement can be provided directly in your text, or as a footnote or an endnote.

The use of Offensive Security trademarks in your domain names is prohibited because such use will lead to the confusion of customers. Any other use outside of the scope of the Trademark Policy is not permitted without express written permission of Offensive Security.

You may make t-shirts, desktop wallpaper, or other merchandise with Offensive Security Marks on them, though only for yourself and your friends (meaning people from whom you don't receive anything of value in return). You can't put the trademarks on anything that you produce commercially (whether or not you make a profit) —

at least not without receiving written permission.

Contact

If you have any questions or comments, or wish to report misuse of the Offensive Security trademarks, please contact us.

Kali Linux Root User Policy

Most distributions encourage their users to use normal user privileges while running the operating system. This is sound security advice, as this behaviour provides an extra layer of security between the user and OS. This is especially true for multiple user systems, where user privilege separation is required.

By nature, Kali Linux is a security and auditing platform, where many tools need to run with root privileges. Generally, when using Kali Linux, being in a multi-user environment is unlikely and therefore the default Kali user is "root". Additionally, [Kali Linux is not recommended for use by Linux beginners](#) who might be more prone to making destructive mistakes while using the super user account.

Penetration Testing Tools Policy

Kali Linux Tools Policy

We realize that there are many tools or scripts that can do the same job. Some are better than others, some are just a matter of personal preference. With this in mind, keeping an updated, useful penetration testing tool repository is a challenging task. The Kali Development team uses some of the following litmus tests to determine whether a specific tool should be included in our Distribution.

- Is the tool useful / functional in a Penetration Testing environment?
- Does the tool contain functionality of other existing tools?
- Does the licensing of the tool allow for free re-distribution?
- How much resources does the tool require? Will it work in a “standard” environment?

Depending on the answers to these questions, and other considerations, we then decide if the tool should be marked for inclusion in Kali.

The majority of the members of the Kali development team are penetration testers, and thus rely on our combined experience to choose the best tools that add the most value to the Kali distribution, while taking in other considerations as well. Tools which are specifically aimed at DOS, DDOS or anonymity are rarely used in legitimate engagements, and are therefore not installed by default in Kali Linux.

New Tool Requests

We are always open to adding new and better tools to our distribution, however a valid case must be made for each tool. Please put some thought and effort into the tool submission and do not just send the developers a one line request. Submissions for new tool requests can be made through our [Kali Linux bug tracker](#).

Kali Network Service Policies

Kali Linux deals with network services differently than most other distributions. Most importantly, Kali does not enable any externally-listening services by default with the goal of minimizing exposure when in a default state.

Default Disallow Policy

Kali Linux will disallow network services to persist across reboots by default. The following example can be seen when attempting to install a tool which would by default would start a network proxy service on TCP port 3142:

```
root@kali:~# apt-get install apt-cacher-ng
...
Setting up apt-cacher-ng (0.7.11-1) ...
update-rc.d: We have no instructions for the apt-cacher-ng init script.
update-rc.d: It looks like a network service, we disable it.
...
root@kali:~#
```

Notice how the update-rc.d script disallowed persistence of the apt-cacher-ng daemon by default.

Service boot persistence

In certain situations, we'll actually want certain services to persist over reboots. To allow for this, we can enable a service to persist through reboots using the update-rc.d script as follows:

```
root@kali:~# update-rc.d apt-cacher-ng enable
update-rc.d: using dependency based boot sequencing
```

Service whitelists and blacklists

Service whitelists and blacklists can be found in the **/usr/sbin/update-rc.d** file. Through this file you can

explicitly allow or deny services to automatically boot in their default state.

```
root@kali:~# tail -95 /usr/sbin/update-rc.d |more  
}
```

```
__DATA__
```

```
#
```

```
# List of blacklisted init scripts
```

```
#
```

```
apache2 disabled
```

```
avahi-daemon disabled
```

```
bluetooth disabled
```

```
cups disabled
```

```
dictd disabled
```

```
ssh disabled
```

```
...
```

```
#
```

```
# List of whitelisted init scripts
```

```
#
```

```
acpid enabled
```

```
acpi-fakekey enabled
```

```
acpi-support enabled
```

```
alsa-utils enabled
```

```
anacron enabled
```

```
...
```

Kali Linux Security Update Policies

Kali Linux is tightly woven with the Debian repositories and as such, receives security updates just as frequently as the main Debian distribution for all the packages that Kali left unchanged (i.e. the vast majority). Other packages are supported on a best-effort basis by the Kali team.