# Table of Contents

**Overview** 3

**Installing Kali** 4
- Download and Copy Image to SD Card 4
- Get a Terminal on Your New Kali Box 5
- Basic Post-Installation Setup 7

**Configuring the PiTFT** 8
- If You Don't Care About Using Kali's Kernel Patches 8
- If You Want the Kali Kernel Patches 9
Overview

Kali Linux is a [distribution](https://adafru.it/CeK) especially aimed at [penetration testing](https://adafru.it/eZN) and network security applications. (It's a successor to [Backtrack Linux](https://adafru.it/eZO).)

Kali isn't intended as a general-purpose desktop OS for end users. Instead, it's a collection of useful tools for monitoring, exploring, and attacking networks. It comes out of the box with tools like [Wireshark](https://adafru.it/eDB), [nmap](https://adafru.it/eZP), and [Aircrack-ng](https://adafru.it/eZQ), and is particularly useful in situations where you just want a disposable machine/installation with some network tools.

Enter the Raspberry Pi: Cheap, portable, low-power, and easy to customize. There's been a lot of interest in using small ARM boxes like the Pi with Kali, and it's well-supported by the maintainers.

Since the Raspberry Pi 2 was released, we've gotten a series of requests for help with getting PiTFT displays to work with Kali on the Pi 2. This guide explains how to do that, and includes a kernel package built with both our PiTFT configuration and the patches applied for a standard Kali Linux build.

It goes without saying that Kali is built on a set of tools that can be used maliciously. Less obviously, they can get you in a lot of hot water.
Three guidelines for using these tools:

1. Be good to other people: Don't violate people's privacy, steal their resources, or break their networks.
2. Do the reading: Learn the purposes and effects of your tools.
3. Remember that even considered, ethical use of the tools on networks you don't own can be received badly by authorities, or violate some broadly-written laws.

If you’re new to topics like pentesting, start by exploring networks you own or have sanctioned access to.

You’ll need the following:

- A Raspberry Pi or Pi 2 ([https://adafru.it/eCB](https://adafru.it/eCB)) (this guide is intended for the Pi 2, but may be of use to others)
- An SD card appropriate for your hardware, 4 gigs or larger ([https://adafru.it/eZR](https://adafru.it/eZR))
- A network connection on the Pi
- PiTFT Plus (best for use with the Pi 2 and Pi A+ or B+) ([https://adafru.it/eZS](https://adafru.it/eZS))
- PiTFT original (best used with the Pi 1 model B) ([https://adafru.it/e27](https://adafru.it/e27))

This guide assumes some experience with GNU/Linux systems ([https://adafru.it/sdn](https://adafru.it/sdn)), and relies heavily on the command line ([https://adafru.it/slE](https://adafru.it/slE)).

## Installing Kali

### Download and Copy Image to SD Card

First, check out the Kali Linux - Raspberry Pi ([https://adafru.it/eZW](https://adafru.it/eZW)) page. You should find a link to a downloads page ([https://adafru.it/eZX](https://adafru.it/eZX)) containing a recent image for
the Raspberry Pi A/B+ or the Pi 2 - grab the one that's appropriate for your hardware. I went with *kali-1.1.0-rpi2.img.xz*.

As of this writing, the v1.0.9 image for the Raspberry Pi A/B+ TFT may work out of the box on a Model B+ with a PiTFT, and might even supply all the drivers and configuration you need for the PiTFT display. It will not boot on a Pi 2, which is why we're going to do it the 'long' way.

The rest of the Kali installation instructions are quite good - you could probably just follow them and come back here.

On my Ubuntu laptop, I decompressed the image with:

```
xz --decompress kali-1.1.0-rpi2.img.xz
```

And used the following `dd` invocation to copy the image to an SD card in my USB card reader:

```
sudo dd if=kali-1.1.0-rpi2.img of=/dev/sdc
```

**Get a Terminal on Your New Kali Box**

If you're using an HDMI monitor and keyboard, you should see a login prompt after the Pi finishes booting, looking something like this:
By default, the user is root and the password is toor.

Alternatively, you can make sure the Pi is connected to the network, and use an SSH client from another machine. That's the approach I'm taking:

```
$ ssh
```

I know that my Pi is at 192.168.1.4 because I configured my router to always assign that IP address to its ethernet interface's MAC. If you don't know yours for sure, you can take a couple of approaches. The simplest is probably to log in from a console and type `ifconfig`:

```
eth0 Link encap:Ethernet HWaddr b2:7e:31:15:91
  inet addr:192.168.1.4 Bcast:192.168.1.255 Mask:255.255.255.0
  inet6 addr: fe80::2b27:17ff:fe73:1591/64 Scope:Link
  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  RX packets:1244 errors:0 dropped:0 overruns:0 frame:0
  TX packets:506 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:80043 (76.6 KiB) TX bytes:77634 (75.2 KiB)
```

Just look for the `eth0` line containing a string like `inet addr:192.168.1.4`.

If that's not an option and you have access to a machine with Nmap (https://adafruit.it/eZP) installed, you can instead try doing a ping sweep of your network with a command like `nmap -sn 192.168.1.1-254`:
You could also use [the Adafruit Pi Finder](https://adafruit.it/jE2) from a Windows, Mac, or Linux desktop to locate the Pi and get a terminal.

## Basic Post-Installation Setup

Once logged in, you should do a bit of simple housekeeping on the new installation.

Since it's not very safe to leave the default password on a machine, start by setting a new password for the root user with the `passwd` command:

```
root@kali:~# passwd
```

Once that's done, reconfigure the OpenSSH server to generate new host keys with `dpkg-reconfigure openssh-server`:

```
root@kali:~# dpkg-reconfigure openssh-server
Creating SSH2 RSA key; this may take some time ...  
Creating SSH2 DSA key; this may take some time ...  
Creating SSH2 ECDSA key; this may take some time ...  
[ ok ] Restarting OpenBSD Secure Shell server: sshd.
```

That should be it! Now we can move on to setting up the PiTFT.
Configuring the PiTFT

For this guide, I'm using the PiTFT - Assembled 320x240 2.8" TFT+Touchscreen (http://adafruit.it/dDE), but these instructions should work for any hardware supported by the Adafruit PiTFT Helper (https://adafruit.it/eln), including the 2.8" capacitive version, the 2.2" version, and the 3.5" version (https://adafruit.it/e27).

If You Don't Care About Using Kali's Kernel Patches

Detailed instructions for configuring a PiTFT on a Raspbian system can be found here:

Adafruit PiTFT - 2.8" Touchscreen Display for Raspberry Pi

https://adafruit.it/eZZ

...and in fact, these would mostly work for Kali. If you just want a working Kali install with the stock Adafruit kernel, you can mount the boot partition, and then follow the instructions for installing adafruit-pitft-helper plus our kernel (https://adafruit.it/sha). Here's the quick and dirty version.

First, while logged in as root, make sure the boot partition (the first partition on the SD card) is available for writing:
mount /dev/mmcblk0p1 /boot

This will let us install a custom kernel and rewrite files like `config.txt` and `cmdline.txt` to configure the display. Next, add `apt.adafruit.com` to your package repository lists, and install `adafruit-pitft-helper`:

```
curl -SLs https://apt.adafruit.com/add | bash
apt-get install -y adafruit-pitft-helper
```

Then wait until everything has installed (this will take quite a while), and run the helper script:

```
adafruit-pitft-helper -u /root/ -t 28r
```

Make sure you change `-t 28r` to an appropriate value for your hardware, if it differs from the 2.8" resistive screen. `adafruit-pitft-helper -h` will list the valid values.

### If You Want the Kali Kernel Patches

This process is a bit more manual, at the moment, but should work.

First, as above, mount the boot partition:

```
mount /dev/mmcblk0p1 /boot
```

Next, you'll need some kernel packages.

If you're a glutton for punishment, you can build these using our [Raspberry Pi Kernel-o-Matic](https://adafruit.it/epp), which uses Vagrant to spin up a little virtual machine just for cross-compiling Pi kernels. We have a [guide](https://adafruit.it/Cfc) for the basics of installation; in particular, here we need to do a custom build ([https://adafruit.it/Cfy](https://adafruit.it/Cfy)) using this branch of the Adafruit kernel repository ([https://adafruit.it/f02](https://adafruit.it/f02)) which contains Kali's patches along with our tweaks for the PiTFT. Once you have the Vagrant box running, do `vagrant ssh` to connect, and then build the kernel like so:

```
sudo adabuild -b rpi-kali-3.18.y
```

If you'd rather skip this process, you can download a pre-built set of packages here:
Download the prebuilt Kernel package
https://adafru.it/f03

or by running

```
wget http://adafruit-download.s3.amazonaws.com/adafruit_pitft_kernel_1.20150420-1.tar.gz
```

Either way, once you have a file like `adafruit_pitft_kernel_1.20150420-1.tar.gz`, copy it to `/root` on your Pi, and extract it like so:

```
tar xf adafruit_pitft_kernel_1.20150420-1.tar.gz
```

Now, run the installer script:

```
cd adafruit_pitft_kernel_1.20150420-1
./install.sh
```

This will take a long time. Once finished, it'll prompt to reboot the Pi immediately. It's safe to say yes, or no if you want to check the contents of `/boot` to make sure that the kernel and related files got replaced properly, which you can do with `ls -l /boot`:

```
root@kali:# ls -l /boot
```

Once you've done that, restart the machine with `reboot`.
Finally, download and run the PiTFT helper script. (Grab it directly from GitHub rather than installing from apt.adafruit.com, because the packaged version depends on a different kernel package.)

```
git clone https://github.com/adafruit/Adafruit-PiTFT-Helper.git
```

Mount the boot partition again:

```
mount /dev/mmcblk0p1 /boot
```

And run:

```
cd Adafruit-PiTFT-Helper
./adafruit-pitft-helper -u /root/ -t 28r
```

You'll be prompted to answer a few questions.
Your output will likely look a bit different from this (I recorded the above GIF after I'd already configured the screen once), but should be similar.

Now **reboot** again, and you should be good to go!

You should also be able to run **startx** at the prompt and land on the desktop.