

How To: Turn Your Linux Box Into a WiFi Hotspot

Have you ever wanted to use your Linux computer as a wireless hotspot? It's actually pretty easy. This article will get you started and it really isn't all that difficult. We will actually be cribbing a bit of this article from the software's homepage, but with some more information given.

Make a Linux WiFi Hotspot

For many years, I used my own router that I had made. It was built on Linux. The preceding version ran on BSD, but that's not important right now. Today, you can get a NUC or Pi for dirt cheap and so making a new router is back on my list of things to do.

All of the varied software and hardware components are already there, but I want to enable wireless connectivity and that's what we're going to look at today. The tool we're going to use is called 'linux-wifi-hotspot' Which is a great tool, complete with GUI if wanted, written by lakinduakash. It has only been around for a few years, but it's spoken of very highly and it just works and works well.

The software is easy enough to install. If you're using Debian/Ubuntu, just add the PPA and install the software. To add the PPA, you just run:

```
[code]sudo add-apt-repository ppa:lakinduakash/lwh[/code]
```

On a modern OS, you shouldn't need to do this, but you might want to go ahead and run a quick update with:

```
[code]sudo apt update[/code]
```

Then you can install the software. To do that, it's just:

```
[code]sudo apt install linux-wifi-hotspot[/code]
```

If you want, you can visit the link above, click on releases, and download the .deb file for the current release and just install it with gdebi.

If you're using Arch (or Arch based distros) it looks like you can just go ahead and install it with:

```
[code]yay -S linux-wifi-hotspot[/code]
```

It should be noted that I did not actually test that very well. I gave it a quick test in Manjaro and it said it couldn't find all the required packages. Manjaro is not Arch, but based on it. I don't have an Arch VM configured without doing some serious digging through my backups, so I am unable to confirm it.

Then, you can go ahead and start it. You can also go ahead and make it start at boot, which would be prudent if you intended to use this to make your own router. It's really self-explanatory and without specific questions for using it, I'm just going to refer you to the man page and the information at the project page.

But, before you can even do all of this, you need to know that your wireless adapter actually supports doing this. To find out, you need to know if your wireless adapter supports "AP" mode. AP obviously meaning 'Access Point'.

To check this, you need to run the following command:

```
[code]iw list | grep AP[/code]
```

The project page is noticeably silent with this, but it's a necessary step. See, you need to know if your hardware actually supports it before you even bother trying. Come to think of it, I probably should have put this closer to the top of the page! I ain't editing that!

Anyhow, the output should contain one or both of the following lines:

Device supports AP scan.

And/Or:

Driver supports full state transitions for AP/GO clients.

So long as you see one or both of those, you should be all set to proceed. If you don't see either of them, there's no software solution and you'll need to get hardware that supports AP mode. In many cases, that'll mean doing a bunch of research and may even mean contacting the vendor or OEM.

Nobody appears to have compiled a list of hardware that supports AP mode and I don't think I've ever bought wireless adapters that explicitly stated they do on the box. As near as I can tell, more modern adapters support it just fine, so you'll probably be alright.

Alright, there's your article for the day. I have no idea if you want to make a WiFi hotspot for your Linux box, but now you know how. Thanks for reading and don't forget to sign up for the newsletter. Also, if you rate the articles I'll be able to see the kind of content you prefer. That'd probably be beneficial.

How To: List USB Device Info From The Terminal (lsusb)

In this article we will learn to use the terminal to check

what USB devices are connected to your computer. We will be using the 'lsusb' command for this one. This article should be both quick and easy.

You may be interested in the three previous articles. In those, we covered 'lshw', 'lscpu', and 'lspci'. Seeing as we've got a good thing going, we'll go ahead and cover 'lsusb' this time. Like the rest, the function becomes evident by the name, and the man page describes it as thus:

lsusb – list USB devices

You shouldn't have to install anything. Your distro should already have the ability to run this command. So, without further ado, let's open your terminal by pressing CTRL + ALT + T. Once you have it open, you can give the command a try.

```
[code]lsusb[/code]
```

This one doesn't need any elevated permissions (see further information below), you can run it as a regular user. As you see from the output, all your connected USB devices should be listed and enumerated. Quite frankly, that's about all you'll ever really need to do with the command – but there is indeed more.

So, what else can you do with it? Well, for starters you can see it in tree view. That may not seem important – but wait for it, there's another use coming up and the tree view is great for seeing what's attached to what hub without digging behind your computer. Let's go ahead and use it with the tree switch:

```
[code]lsusb -t[/code]
```

That's easy enough and informative enough. Like I said, this is useful if you have USB hubs and want to see what's connected to what. But, that's still not the neatest thing you can do with 'lsusb'. Let's go back to the original command.

```
[code]lsusb[/code]
```

Take a look at the output. For example, you might see something like this:

```
[code]Bus 001 Device 005: ID 0bda:8178 Realtek Semiconductor Corp. RTL8192CU 802.11n WLAN Adapter[/code]
```

Now, I want you to look at those first two sections, 'bus' and 'device'. Those are actually pretty useful. You just need to know how to use them. Keep them in the same order, and add them to the following command:

```
[code]lsusb -D /dev/bus/usb/<bus #>/<device #>[/code]
```

So, using my example above, you end up with a command that looks like:

```
[code]lsusb -D /dev/bus/usb/001/005[/code]
```

See, the -D switch is for Device and, curiously, the manual says that you need to be root to use it. However, I've not ever needed root to do so. Allow me to quote again:

*Do not scan the /dev/bus/usb directory, instead display only information about the device whose device file is given. The device file should be something like /dev/bus/usb/001/001. This option displays detailed information like the v option; **you must be root to do this.***

Try as I might, I can't see a lick of difference when I run the command without root. So, I have no idea why the manual says I should, or that I must. If anyone knows why, please leave a comment. (Contrary to popular opinion, I do not in fact know everything!)

Anyhow, that last use of lsusb is the most informative. The output is too long for me to bother copy/pasting here, though padding the article would be nice. So, go ahead and run it for

yourself. All you need to do is check the bus and device numbers, insert them properly into the path, and you'll get a ton of info you probably didn't know was available.

Once again, thanks for reading. Be sure to scroll up, look right, and sign up for the newsletter. You'll get a lovely little notification every time there's an article published. Feel free to leave a comment and let me know why it says root is required. As near as I can tell, that isn't necessary?

How To: List PCI Information From The Terminal (`lspci`)

Continuing with a theme, this article will explain how to use the terminal to view what's attached via PCI (Peripheral Component Interconnect) to your computer. PCI devices are the ones in the add-on slots in your computer. They're typically internal devices, though you can get USB powered external devices with PCI-e slots, they kind of defeat the purpose. For this article, we'll be using `lspci`.

You may also be interested in reading up about `lshw` and `lscpu`. I didn't intend for this to be a 'series' but that's what it's starting to look like. That's nice, however. They're small bites that let you sample the buffet that is the Linux terminal.

This one is just a little more complicated than the last. For starters, you may not have `lspci` immediately available. You may have it installed by default, but you might not. It really depends on your distro. Either way, it's in your default repositories.

If you don't have it installed, it's in the package 'pciutils'. So, use your package manager to install it. For example, with apt you'd do this:

```
[code]sudo apt install pciutils[/code]
```

Once you have it installed, you'll see the manual defines 'lspci' as:

lspci – list all PCI devices

If you don't know, PCI devices are a class of devices that are add-ons to your motherboard. They're mostly the devices that go in the slots, such as graphics cards or sound cards. They can be used for all sorts of things, these days even being used for M.2 devices that hold SSD drives for rapid storage. They're great because they have a fast bus speed, meaning that your computer can interact with them faster because data moves faster in both directions than it does for, for example, a USB device.

Except PCI devices aren't just limited to the things that go in slots. Your motherboard probably uses that same spec to interact with other devices. For example, your Ethernet and sound card may be listed – even though they're 'on-board' and not actually add-on cards. That's your motherboard using the same sort of bus spec but not actually using a physical port. Additionally, there are different types of PCI specifications, but we don't need to get into that today.

And, well, once again the 'lspci' name tells you what it does, now that you know that's what it does. It lists PCI devices, just as the name implies. There are a couple of ways that you'd realistically want to use it.

NOTE: The results from 'lspci' are drawn from the The PCI ID Repository and may actually not be accurate. Yup. You could get inaccurate results from this command, but we throw it

around daily as though it's infallible. And now you know...

Where were we? Oh, yeah... We were going to use 'lscpi' for something useful. So, let's crack open that terminal by using your keyboard to press CTRL + ALT + T and we'll first enter:

```
[code]lspci[/code]
```

That will list all your PCI devices in a quick and easy to read list. You may also want to get the verbose output and that's done like:

```
[code]lspci -v[/code]
```

The output of that command should be fairly obvious. After all, it does what it says it does on the tin. That's also perhaps the more useful of the commands. It's definitely the one that I use most frequently.

If you take the numbers from the start of each line from the output of the above command and use the -t switch you'll actually get an understandable 'tree' output that will help you further understand what's going on inside your case without opening it up. If one PCI device has multiple entries (as many do) it'll make that easier to understand. It's simply:

```
[code]lspci -t[/code]
```

However you can easily put the two of those together and simply get a great verbose tree output with:

```
[code]lspci -vt[/code]
```

That's plenty easy to understand but some folks may find it a bit overwhelming. I don't usually need that much information, so I tend to run the command without any switches. As I bounce between devices, it's enough to just check and make sure I know what I'm working with.

NOTE: Older versions required `-vvv` for verbose and `-tree` were needed to perform those operations. The current versions simply use the `-v` and `-t` switches.

And there you have it. Yet another way to view hardware information from within the terminal. You may have noticed a trend and probably can narrow down your guess as to what the next article is gonna be about! If `lscpi`'s parent package is not installed, it's really easy to get and you can then run the command. If you don't have access to many tools, you almost certainly have access to this one.

Like always, thanks for reading. Be sure to sign up for the newsletter. I'll only send you site-related material and won't sell your email address to anyone. I promise, I won't ever send you any spam – just site stuff!

How To: List CPU Information From The Terminal (`lscpu`)

In the last article I wrote, I explained how to use `'lshw'` to get information about your hardware from the terminal. This article will show you how to gather CPU information from the terminal.

This will be yet another quick article, but again this is a valuable tool to do hardware diagnostics when you don't have other tools available.

Just like last time, the command is self-evident once you really look at it. The command is `'lscpu'` and it does what you'd expect – `'list CPU information'`. It is defined as thus in the manual:

lscpu – display information about the CPU architecture

So, let's give it a shot. Crack open your default terminal emulator by pressing CTRL + ALT + T. And, let's just start by entering:

```
[code]lscpu[/code]
```

And, there you have some fairly easy to understand. It's often used as a quick way to tell if your CPU supports 64 bit instructions. You'll see something like this if it does:

```
[code]Architecture: x86_64[/code]
```

NOTE: Pretty much all modern computer hardware supports the 64 bit instruction set, but there are still some 32 bit machines out there and being used.

Unlike the 'lshw' command, this one doesn't need to be run as an administrator. It runs as a regular user just fine and running it as an admin doesn't get you any more information.

Also unlike the 'lshw' command, there isn't any other way to run it that's all that interesting. You can read the man page, but you'll seldom need to use this in any other way other than the way described here. Just type the command, get the information, and move on with whatever it was you were doing!

See? I told you this one would be another quick and easy article. Thanks for reading and don't be scared of signing up to the newsletter. It's not like you'll be inundated with piles of unwanted email! You'll just get notified when there's a new article and I promise to not sell your email address to anyone.

How To: List Hardware From The Terminal (lshw)

You may not always have inxi available. The person trying to help you may want more specific info. You may need more information about your hardware. Who knows? There's all sorts of reasons to use 'lshw' in your day-to-day computing.

Today is another short and easy article (you're welcome). It's just a very brief command, one of several, that we're going to learn how to use. The command in question is 'lshw'. The manual helpfully defines it as:

lshw – list hardware

Which, now that you look at it, makes perfectly good sense. This is one of those Linux commands that's not at all cryptic when you think about it. It looks like what it does, it does what you'd intuitively think it does. (It's actually a subset of the info you get with 'hwinfo', but that's not important right now!)

What is important is that there are two realistic ways to use it – and both of them should be run as an administrator. The first is just as you'd expect:

```
[code]sudo lshw[/code]
```

That will output a ton of information about the hardware you have in (or connected to) your computer. It's rather overwhelming and it's not often that you'll be asked to post all of it. You may be asked to (or want to) run it with the -C option, such as:

```
[code]sudo lshw -C cpu[/code]
```

You may even be asked to use grep with it, such as:

```
[code]sudo lshw | grep wireless[/code]
```

Then, there's one more way to run the program (under normal circumstances) and that's to run the whole thing and to get the whole output in a shorter format. The command is probably just like you'd expect.

```
[code]sudo lshw -short[/code]
```

Lo and behold! Would you look at that! Ha! Isn't that fantastic? You actually get readable output that's suitable for your normal user who just wants to know what sorta hardware they're working with while getting a few extra bits of info! It even includes juicy nuggets like:

```
[code]/1 wlxe4beed0e5f5c network Wireless interface[/code]
```

Now, if you want to refine it even further, why not try this:

```
[code]sudo lshw -short | grep network[/code]
```

And, there you have it... You have another way to see your hardware in the terminal and it should be consistent across any major distro you touch. You shouldn't have any trouble using the command and it's easy to remember when you want to 'list the hardware'. There are other uses, but those two are the most common. For more information about the lshw command try:

```
[code]man lshw[/code]
```

or

```
[code]info lshw[/code]
```

Like always, thanks for reading. Feel free to subscribe to the newsletter. I promise to not give away your private

information – and you never know what sorta incentives I may stick in there. See? I told you this one would be short and easy!