

How To: Generate Sufficiently Complex Passwords In The Terminal

In this article, I will tell you how I use [pwgen](#) to generate random passwords. It is a simple command, but comes in surprisingly handy. This being Linux, there are surely other ways to accomplish this task, but this is the method that I use.

I've been working on a router for the past few weeks. I've been working on getting this site up and running. I've also been working on a new computer that's just meant for testing.

I don't do a whole lot of password sharing and remembering. That means when I need to login to an account I'll often just hit the button to recover my password and then use a new password entirely. This means I've had to generate quite a few passwords. The tool I use to do this is known as *pwgen*.

GENERATE COMPLEX PASSWORDS

The tool 'pwgen' may well be installed by default. If not, it's very likely to be in your default repositories. I checked everything from Fedora to Mandriva, and it looks like you shouldn't have any issues installing it. If you're using a Debian/Ubuntu/derivative and have access to apt, then you can just install with:

```
[crayon-6107f2289aec2767334689/]
```

Once installed, you can just run the 'pwgen' command from the terminal and you'll get a bunch of random passwords. They may, or may not, be sufficiently complex and able to satisfy the various password requirements. It tosses out a lot of passwords, so it's not like you won't have choices.

Me? I use it a bit differently. In fact, you might even say that I use it 'wrong'. After all, the man page describes the pwgen application as:

pwgen – generate pronounceable passwords

I don't actually want pronounceable passwords. I want suitably complex passwords! Fortunately, pwgen can be bashed into shape and can provide all the complex passwords you desire. Here's the command I use:

```
[crayon-6107f2289aeca182667304/]
```

So, what's going on with that? What do those switches do?

- s – random and hard
- n – adds numbers
- y – adds symbols
- c – at least one capital letter
- B – removes ambiguity
- l – one password per line

The 12 is how many characters each password will have and the 6 is how many passwords it will spit out for you. So, for example, you'll get an output like this:

```
[crayon-6107f2289aecc115271121/]
```

In my case, this is already aliased to 'newpass' and that means I don't need to type the entire command when I want a new password. Alas, I've not yet covered aliases. It's on my list of articles to cover, but it's not done yet.

Anyhow, thanks for reading. As you can see, this is the very first article at our new home. I'll slowly move the other articles over here, editing them to match the new format and maybe making some of them a bit more thorough. This site should be faster, have the features that were used at the previous site, and a bit easier to maintain. There's even a

published privacy policy.

Don't forget to sign up for the newsletter. It'll only send out emails when an article has been published and won't include any spam. I won't give your email address out, nor will I sell or trade it.