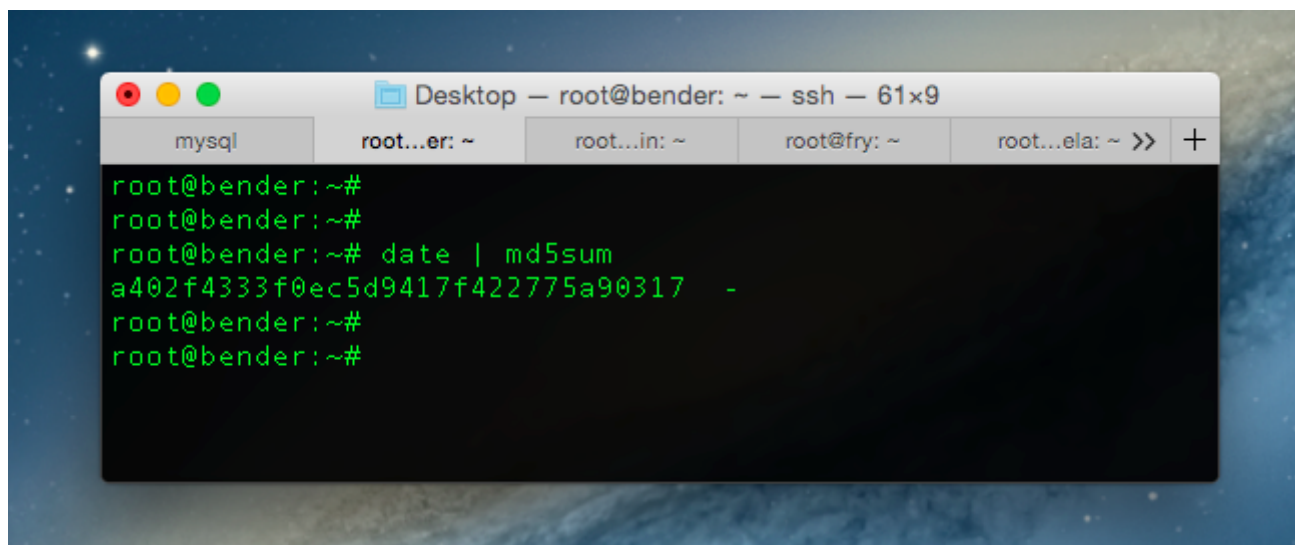


10 Ways to Generate a Random Password from the Linux Command Line

by **Lowell Heddings** on December 20th, 2014

A screenshot of a terminal window titled "Desktop — root@bender: ~ — ssh — 61x9". The terminal shows a series of prompts and commands. The first three prompts are "root@bender:~#". The fourth prompt is followed by the command "date | md5sum", which outputs "a402f4333f0ec5d9417f422775a90317 -". The fifth and sixth prompts are "root@bender:~#".

```
root@bender:~#  
root@bender:~#  
root@bender:~# date | md5sum  
a402f4333f0ec5d9417f422775a90317 -  
root@bender:~#  
root@bender:~#
```

One of the great things about Linux is that you can do the same thing hundreds of different ways—even something as simple as generating a random password can be accomplished with dozens of different commands. Here's 10 ways you can do it.

We gathered all of these commands from [Command-Line Fu](#) and tested them out on our own Linux PC to make sure they work. You should be able to use at least some of these on Windows with [Cygwin](#) installed, though we didn't test all of them—the last one definitely works though.

Generate a Random Password

For any of these random password commands, you can either modify them to output a different password length, or you can just use the first x characters of the generated password if you don't want such a long password. Hopefully you're using a password manager like [LastPass](#) anyway so you don't need to memorize them.

This method uses SHA to hash the date, runs through base64, and then outputs the top 32 characters.

```
date +%s | sha256sum | base64 | head -c 32 ; echo
```

This method used the built-in /dev/urandom feature, and filters out only characters that you would normally use in a password. Then it outputs the top 32.

```
< /dev/urandom tr -dc _A-Z-a-z-0-9 | head -c${1:-32};echo;
```

This one uses openssl's rand function, which may not be installed on your system. Good thing there's lots of other examples, right?

```
openssl rand -base64 32
```

This one works a lot like the other urandom one, but just does the work in reverse. Bash is very powerful!

```
tr -cd '[:alnum:]' < /dev/urandom | fold -w30 | head -n1
```

Here's another example that filters using the strings command, which outputs printable strings from a file, which in this case is the urandom feature.

```
strings /dev/urandom | grep -o '[:alnum:]' | head -n 30 | tr -d '\n'
```



Here's an even simpler version of the urandom one.

```
< /dev/urandom tr -dc _A-Z-a-z-0-9 | head -c6
```

This one manages to use the very useful `dd` command.

```
dd if=/dev/urandom bs=1 count=32 2>/dev/null | base64 -w 0 | rev | c
```

You can even create a random left-hand password, which would let you type your password with one hand.

```
</dev/urandom tr -dc '12345!@#$%qwertQWERTasdfgASDFGzxcvbZXCVB' | he
```

If you're going to be using this all the time, it's probably a better idea to put it into a function. In this case, once you run the command once, you'll be able to use *randpw* anytime you want to generate a random password. You'd probably want to put this into your `~/.bashrc` file.

```
randpw(){ < /dev/urandom tr -dc _A-Z-a-z-0-9 | head -c${1:-16};echo;
```

You can use this same syntax to make any of these into a function—just replace everything inside the { }

And here's the easiest way to make a password from the command line, which works in Linux, Windows with Cygwin, and probably Mac OS X. I'm sure that some people will complain that it's not as random as some of the other options, but honestly, it's random enough if you're going to be using the whole thing.

```
date | md5sum
```

Yeah, that's even easy enough to remember.

There's loads of other ways that you can create a random password from the command line in Linux—for instance, the `mkpasswd` command, which can actually assign the password to a Linux user account. So what's your favorite way?

SHARE:



COMMENTS:



Lowell Heddings is the founder and Editor-in-Chief of How-To Geek. He spends all his free time making sure this site can bring you fresh geekery on a daily basis, and has been doing so for over eleven years.

DID YOU KNOW

A beaver can hold its breath under water for an impressive 10 – 15 minutes (the average human can hold their breath for around 1 minute).

BEST OF HOW-TO GEEK



How to Use All of Windows 10's Backup and Recovery Tools



How to Remap Xbox, PlayStation, and Other Controller Buttons in Steam



How to Mirror Your Windows or Android Device's Screen on Your Roku



How to View (and Monitor) Your Credit Report For Free



7 Ways To Free Up Hard Disk Space On Windows



Nest vs. Ecobee3 vs. Honeywell Lyric: Which Smart Thermostat Should You Buy?



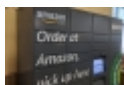
How to Remove a Password From a PDF File



What's the Best Way to Back Up My Computer?



What Is Wake-on-LAN, and How Do I Enable It?



Using Amazon Locker Is Super Easy, and Can Save You Time and Money



GET THE HTG NEWSLETTER

Join 100,000 other subscribers

FOLLOW HOW-TO GEEK

160K+

50K+

90K+

MORE ARTICLES YOU MIGHT LIKE

ABOUT

About Us
Contact Us
Discussion Forum
Review Geek
How-To Geek Pro
Advertising
Privacy Policy
Terms of Use

GET ARTICLES BY EMAIL

Enter your email address to get our daily newsletter.

GET ARTICLES BY EMAIL

FOLLOW US

TWITTER

GOOGLE+

FACEBOOK

RSS FEED

Disclaimer: Most of the pages on the internet include affiliate links, including some on this site.

