USN-4759-1: GLib vulnerabilities

Krzesimir Nowak discovered that GLib incorrectly handled certain large
buffers. A remote attacker could use this issue to cause applications
linked to GLib to crash, resulting in a denial of service, or possibly
execute arbitrary code. (CVE-2021-27218)
Kevin Backhouse discovered that GLib incorrectly handled certain memory
allocations. A remote attacker could use this issue to cause applications
linked to GLib to crash, resulting in a denial of service, or possibly
execute arbitrary code. (CVE-2021-27219)

USN-4733-2: GNOME Autoar regression

USN-4733-1 fixed a vulnerability in GNOME Autoar. The upstream fix
introduced a regression when extracting archives containing directories.
This update fixes the problem.
Original advisory details:

Yiğit Can Yılmaz discovered that GNOME Autoar could extract files outside
of the intended directory. If a user were tricked into extracting a

specially crafted archive, a remote attacker could create files in
arbitrary locations, possibly leading to code execution.

Let’s Count the Installed Packages!

Have you ever wanted to know how many packages you’ve installed? It’s actually a really simple process. I’ll show you!

This is only useful if your distro uses ‘dpkg’, which is actually quite a few distros. If you remember correctly, I’ve previously told you how to use ‘dpkg’ to get a list of installed applications. It was a pretty simple command.

Well, so isn’t this a simple command.

Like pretty much every time, let’s get that terminal opened up. (CTRL + ALT + T)

Now, if you’d followed the above link’s directions, you’d have a file called ‘installed_apps.txt’ and it would be in your Documents directory. If you open that file, you’ll see that not all the lines are actually installed applications. There are some lines that contain data that isn’t an installed application.

But, if you look carefully, you’ll see that the lines with the applications all begin with ‘ii’. We can use that to make our work more accurate.

Now, you could try this:

[code]cat Documents/installed_apps.txt | grep ii | wc -l[/code]

If you’re curious, ‘cat‘ stands for concatenate. It has been around since pretty much Unix v. 1. It basically reads a file and spits out the content. The man page describes it as thus:

> *cat – concatenate files and print on the standard output*

See? Pretty simple.

Next, you’re telling it which file to work on (installed_apps.txt in the Documents directory). Then you’re telling it to look for the letters ‘ii’ and to count the lines that contain them.

Note: In some cases, this will not be 100% accurate. If you have something installed with ‘ii’ in the name then it will count that as well. However, the goal here isn’t actually 100% accuracy, the goal here is to help you get familiar with some of these terminal commands.

So, what if you haven’t followed along and don’t actually have that file? Can you still do this? Absolutely! Watch this:

[code]dpkg -l | grep ii | wc -l[/code]

To be a bit more clear, that little ‘|’ character is called a pipe. You’ll see it fairly often. It’s used to take the commands from one command and use them in another. It goes back to the philosophy of ‘hiding the internals’, with the goal being simplicity and clarity. But, you never have to make the text file to perform this counting exercise.

Again, this command will give you an inaccurate result if you happen to have an application that has an ‘ii’ in the name. That’s fine. This is great for estimation and you really don’t need a hard number for anything. In a quick look, I have exactly zero apps with ‘ii’ in the name. So, in my case the count should be spot on. The goal is to help you get more comfortable in the terminal and get used to some of these commands. They're surprisingly useful, even in day-to-day operations.

Like always, thanks for reading. If you want, you can sign up for the email newsletter. It's over there, in the right sidebar. I promise, I won't send you any spam, I won't sell your email address, I won't give your email address away, and I won't send you pics of my dinner. So, sign up and be the first person on the block to read new articles!

How To: Make Ubuntu Show Asterisks When Typing Password

By default, Ubuntu doesn't show anything when you enter your password in the terminal. This is for security reasons. Someone shoulder-surfing won't be able to see the number of characters in your password. This is how to get some feedback when you enter your password in the terminal.

This one is pretty easy and shouldn't take very long. First, let's open your terminal. Press CTRL + ALT + T and your default terminal should open. Yay!

Now, we need to edit the sudoers file. It's done like this:

```
[code]sudo nano /etc/sudoers[/code]
```

Enter your password and hit enter, of course. (This will be the last time you enter your password in the terminal without some sort of visual feedback!)

Now it gets a little tricky.

Use the down arrow until your at the start of the line that says:

[code]Defaults   mail_badpass[/code]

Press the ENTER button. This should move that line down and leave a blank line above it. Use the arrow button to move up to that blank line and enter:

[code]Defaults[/code]

Then press the TAB button on your keyboard. This will move the cursor to the right location. Add this text:

[code]pwfeedback[/code]

The entire line should look something like:

[code]Defaults   pwfeedback[/code]

Note: This spacing isn't really required so much as it is done for convention and to aid in ease of reading/processing information-dense text more accurately and swiftly. If you want to be diligent, you can even leave a comment, prefaced with a **#**, remarking that you made a change and why you made a change. Comments should be on their own lines.

Anyhow…

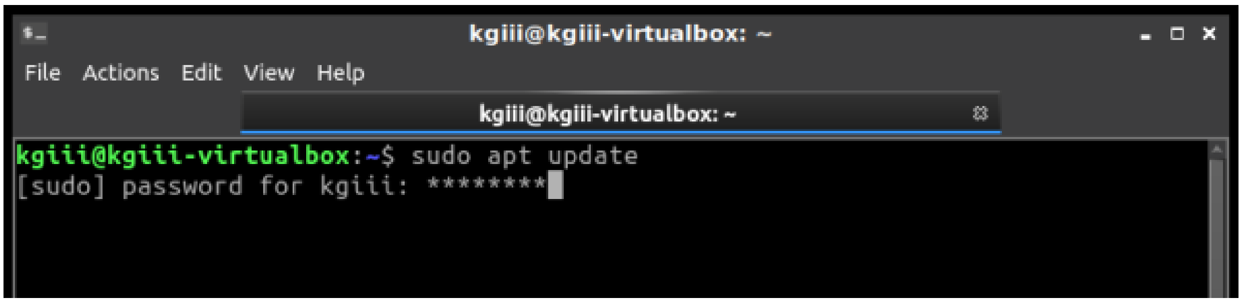
Now, you simply need to save the file. If you've been following along, you'll already know how to do that. If not, here it is again:

Press CTRL + X, then Y, and then ENTER.

Congrats, you're done! You may need to close and reopen your terminal to notice the difference. Test it by opening a new terminal window and tying in:

[code]sudo apt update[/code]

Type your password when prompted and you'll hopefully see some asterisks as feedback. It should look a little like this:


kgiii@kgiii-virtualbox: ~
File Actions Edit View Help
kgiii@kgiii-virtualbox: ~

```
kgiii@kgiii-virtualbox:~$ sudo apt update
[sudo] password for kgiii: ********
```

See? Asterisks for feedback in the terminal.

As always, thanks for reading. Feel free to sign up for the newsletter. The only emails you'll get are notifications when there are new articles. I won't even send you any spam, I pinky swear! Also, I think I'm going to settle on the Helvetica font. It's pretty clear, easy to read, and easy to distinguish numerals from alphabetical characters. I should probably go back through my old articles and make this consistent, but it's too much fun writing new articles!

USN-4757-2: wpa_supplicant and hostapd vulnerability

USN-4757-1 fixed a vulnerability in wpa_supplicant and hostapd. This update
provides the corresponding update for Ubuntu 14.04 ESM.
Original advisory details:

It was discovered that wpa_supplicant did not properly handle

P2P
(Wi-Fi Direct) provision discovery requests in some situations. A
physically proximate attacker could use this to cause a denial of service
or possibly execute arbitrary code.