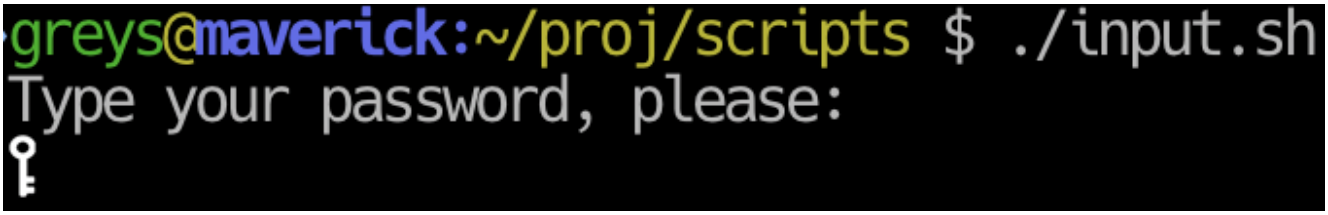
Secure Keyboard Input in bash Scripts



Secure keyboard input in bash

I needed to create a very simple **bash** script that required a password to be provided using keyboard and then used further in the script. Turns out, there's just a way to do it with the **bash** built-in function called **read**.

Standard user input in bash

Here's how a normal user input works: you invoke read function, pass it a variable name. A user is prompted for input by the bash script, and when input is provided it's shown (echoed) back into terminal – so you can see what you type.

First, let's create the script:

$ **vi input.sh**

this will be the content for our file:

```
#!/bin/bash
echo "Type your password, please:"
read PASS
echo "You just typed: $PASS"
```

Save the file (press Esc, then type :wq) and make it executable:

```
$ chmod a+rx input.sh
```

Now we can run the script and see how it works:

```
$ ./input.sh
Type your password, please:
mypass
You just typed: mypass
```

Works great, but seeing the typed password is not ideal. In a real world example I wouldn’t be printing the password back either.

Secure keyboard input in bash

Turns out, read function supports this scenario – just update the script to this:

```
read -s PASS
```

-s is obviously short for secure.

Save the script and run it again, this time typing will not show, but later command should output our input just fine:

```
$ ./input.sh
Type your password, please:
You just typed: mypass
```

Pretty cool, huh?

See Also

- Bash scripts
- Math in bash scripts