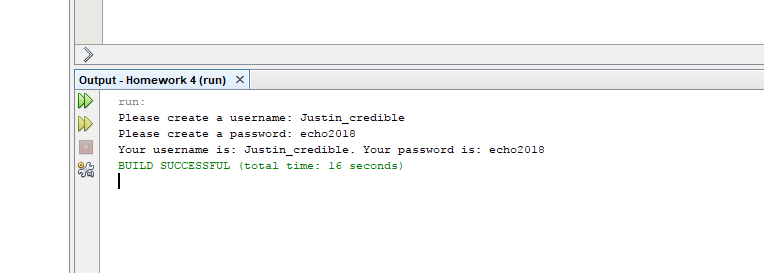
Missing Encryption of Sensitive Data

Encryption of sensitive data, such as passwords, PINs, and secret information, is essential. Whenever sensitive data is being stored or transmitted outside of your control, hackers will be looking for ways to get it. It is almost certain that malicious users will gain access to the content. They could look through your file system, sniff packets, or read your database. If sensitive data is leaked it can be a huge headache for a company.

For our examples, we will have a part of a larger program that asks a new user to create a username and password. We will take a look at a vulnerable version and a mitigated version.

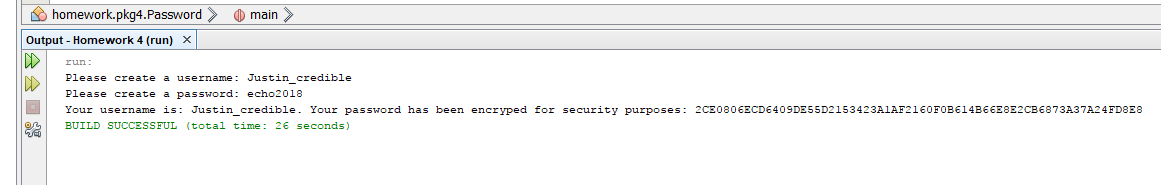
Vulnerable-

Our program asks a user to create a username and password. It then displays the new user’s username and password after storing them in variables. When we print them to the screen, they appear without any problems. This leaves us open to many different angles of attack.



Mitigated-

We will ask the new user for their username and password. We will then store the password in a cryptographic hash version of itself, which we would later store in our database. With all of the random characters, hackers would have an extremely difficult, if not impossible time trying to figure out the input was that gave us this output.

Looking at our code, we can see that when we take the user’s input and store it in a variable called “pw”. We then take that variable and encrypt it using the SHA-256 message digest of the password. It is now a bunch of random characters, which mean nothing to us. However, the program can later decrypt the password using a key and know exactly what the password is. Storing passwords, and even usernames, is much safer with encryption.