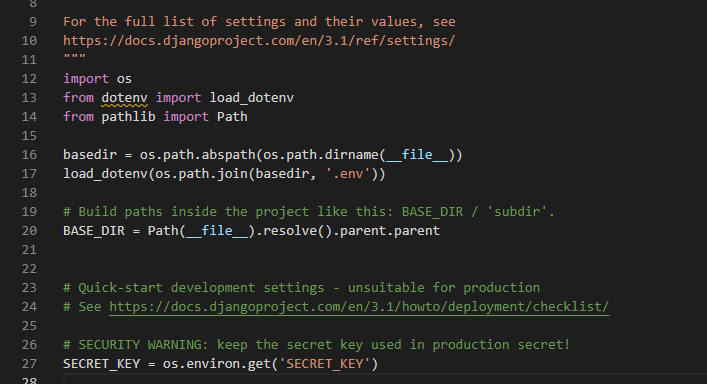
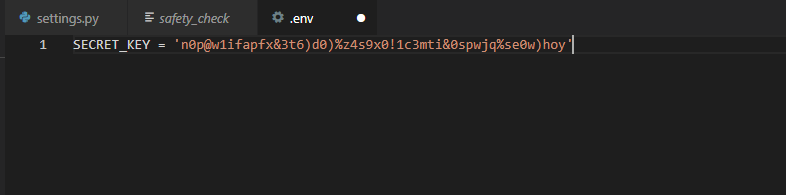
**Secure Coding Evidence**

In order to make sure that any web application is secure. Any secrets and hard coded variables required for things like access and authentication etc. should be moved outside of the actual code files for the applications. These can be stored in environment variables and accessed via correctly coded links and imports.

The below example shows that the secret key has been stored in an external file which is loaded into the environment variables:



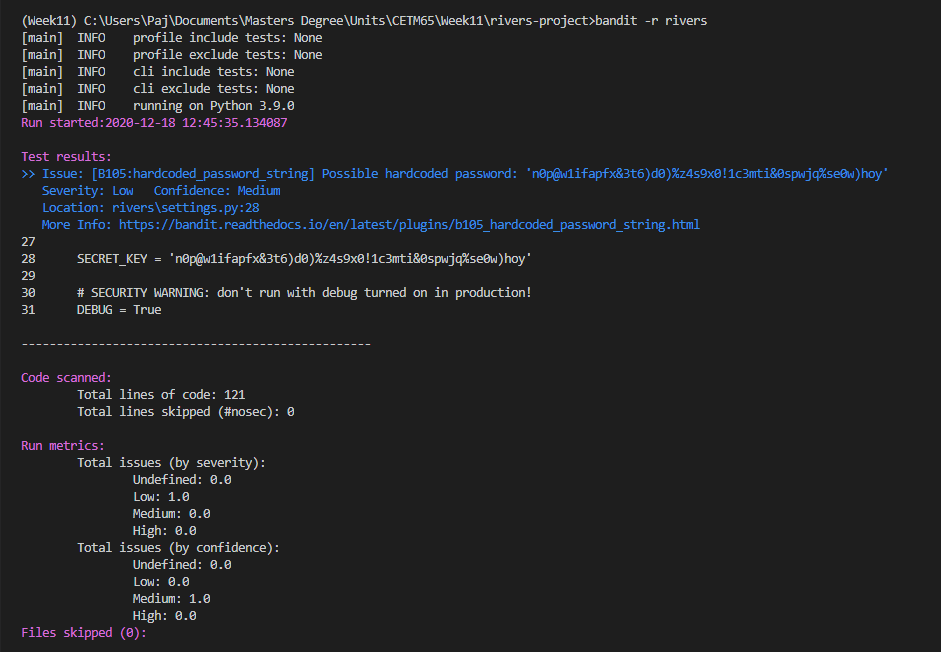
The variables are held in a ‘.env’ file which is not pushed to the repository, instead they are held securely in a vault space etc.



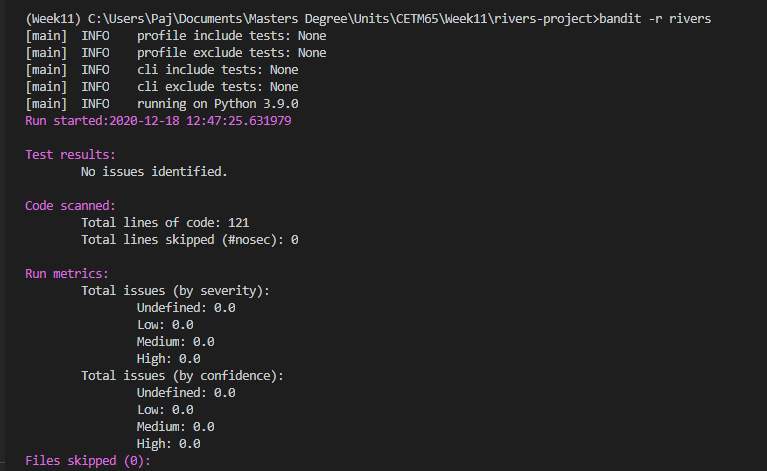
Bandit and Safe Space

In addition, there is a chance that your code either uses packages/libraries that aren’t secure, or has code that is not secure. Usually insecure code has things like hardcoded SQL commands etc., that can be exploited as attack vectors.

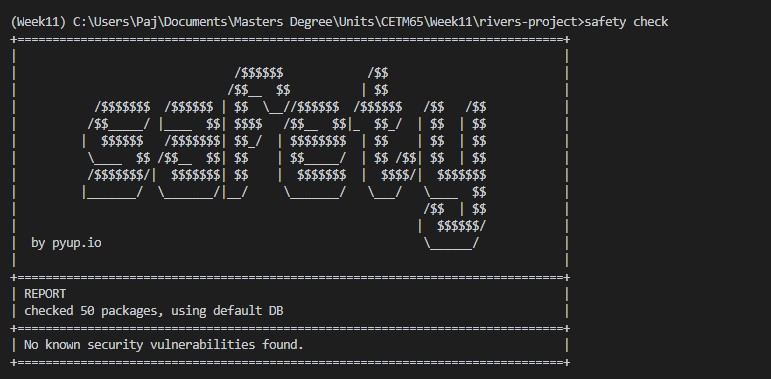
In order to help prevent this, bandit and safe space are 2 packages that can be utilised on any python code to check for the above errors. Bandit will check for any hardcoded SQL etc. and safe space will check that any installed packages are also safe.



In the above example, Bandit has detected that the secret key is hardcoded into the setting.py file and has highlighted this as an issue. After correcting this (using environment variables etc.) we can see that bandit has found no issues or areas of concern.



Safety will check the downloaded and installed packages (installed via pip) and will alert you to any packages that are suspicious or have vulnerabilities.



We can see that there were no security vulnerabilities found in the 50 packages installed on the virtual environment.