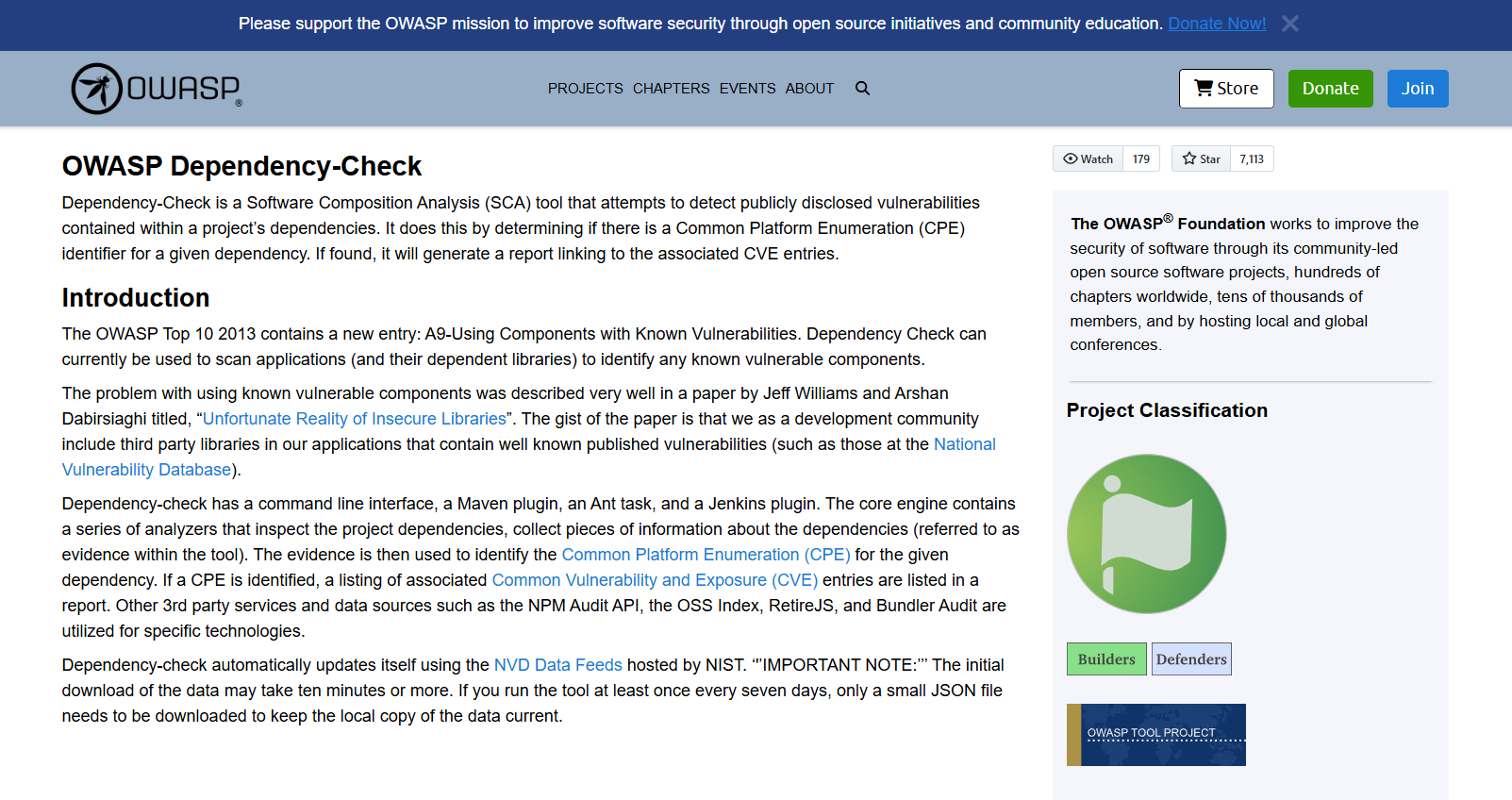
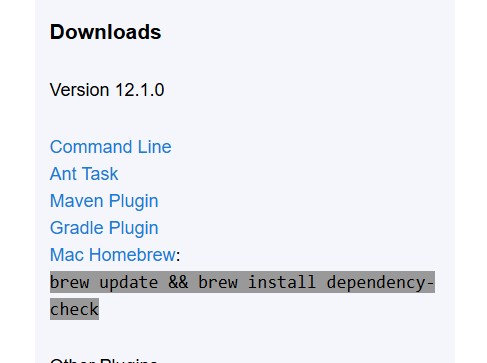
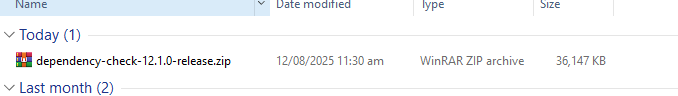
STEP 1



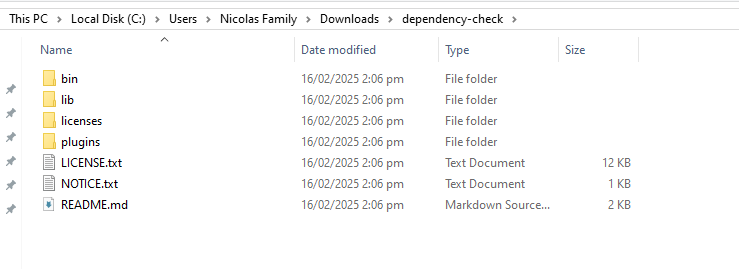
STEP 2



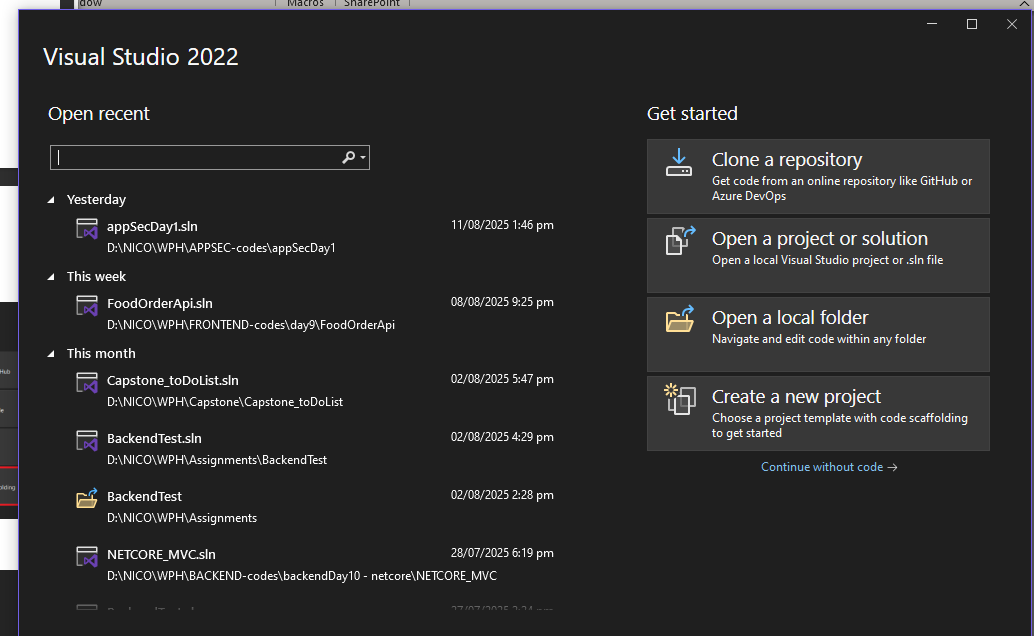
STEP 3



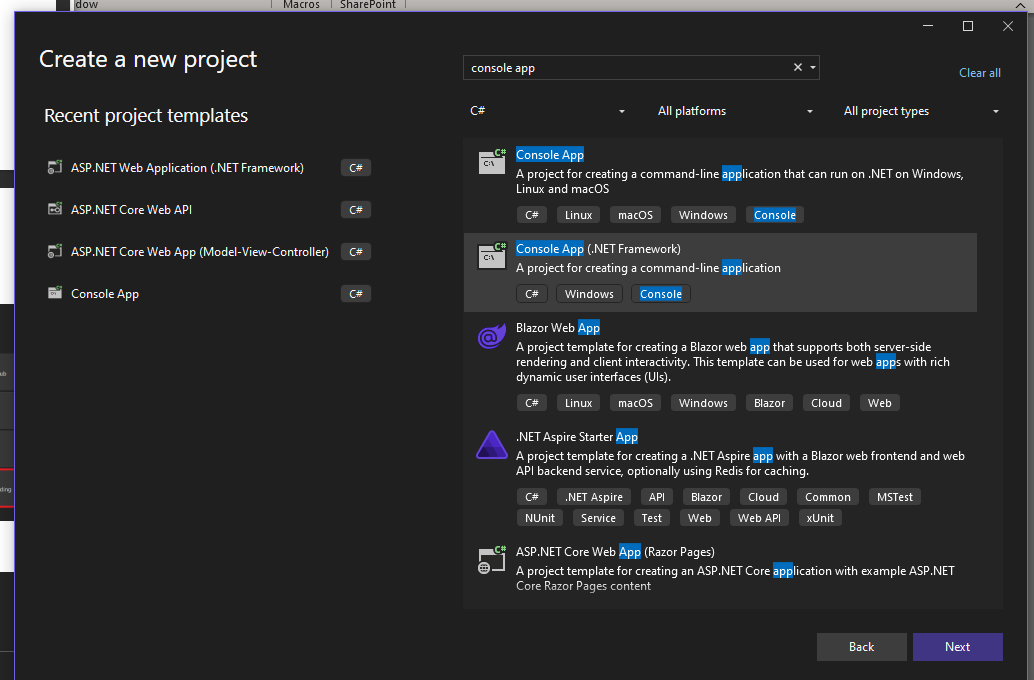
STEP 4



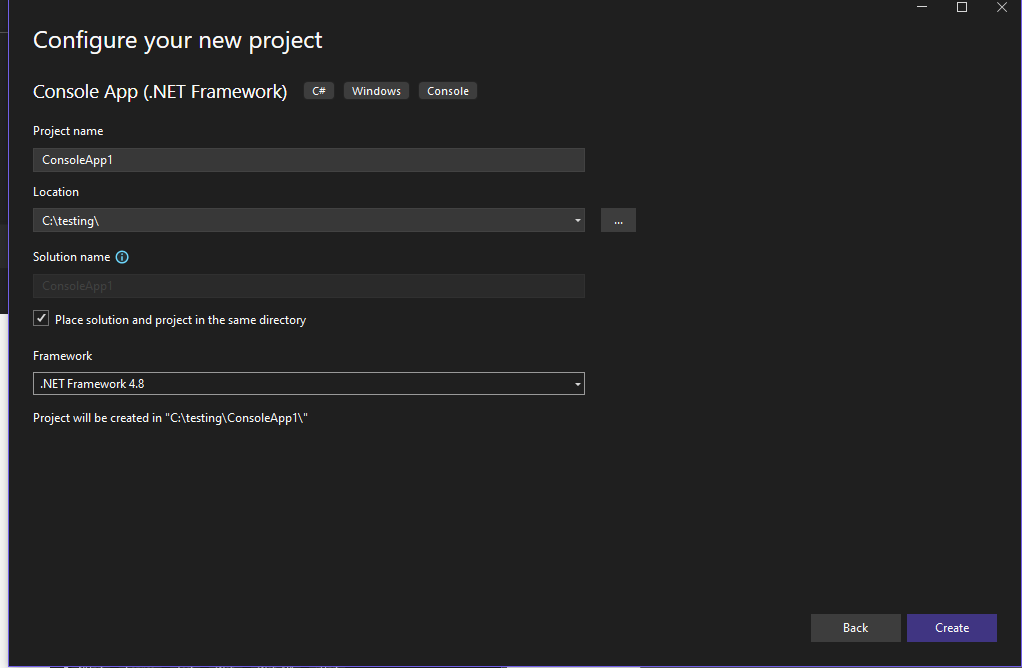
STEP 5



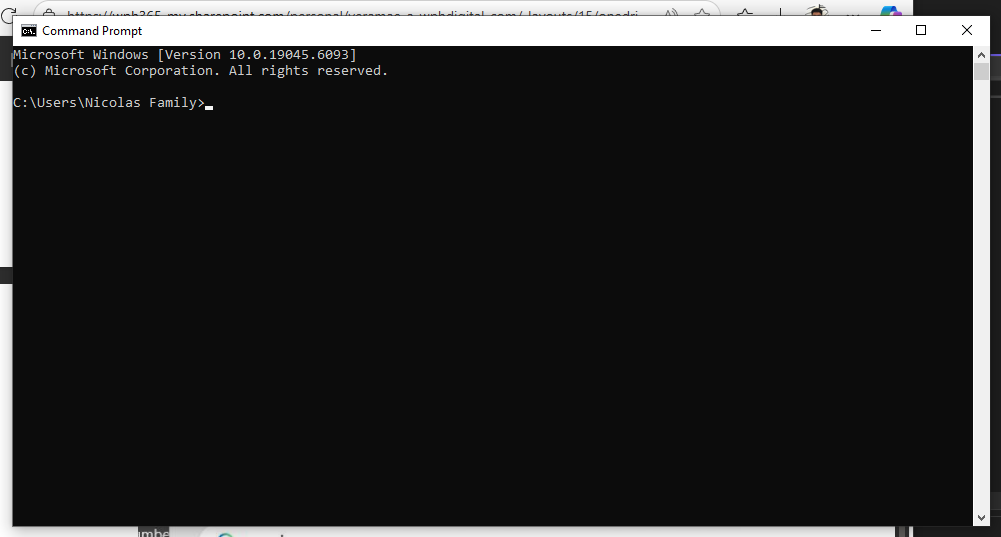
STEP 6



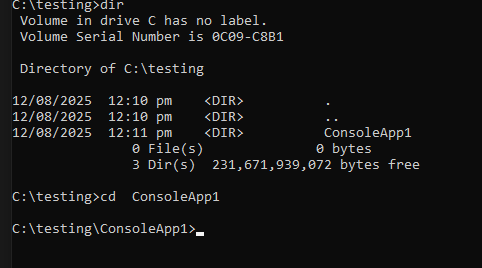
STEP 7



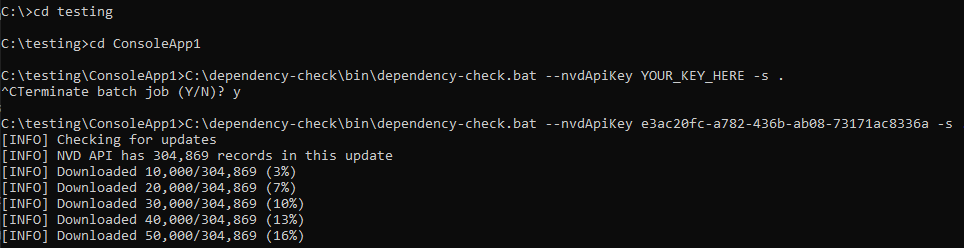
STEP 8

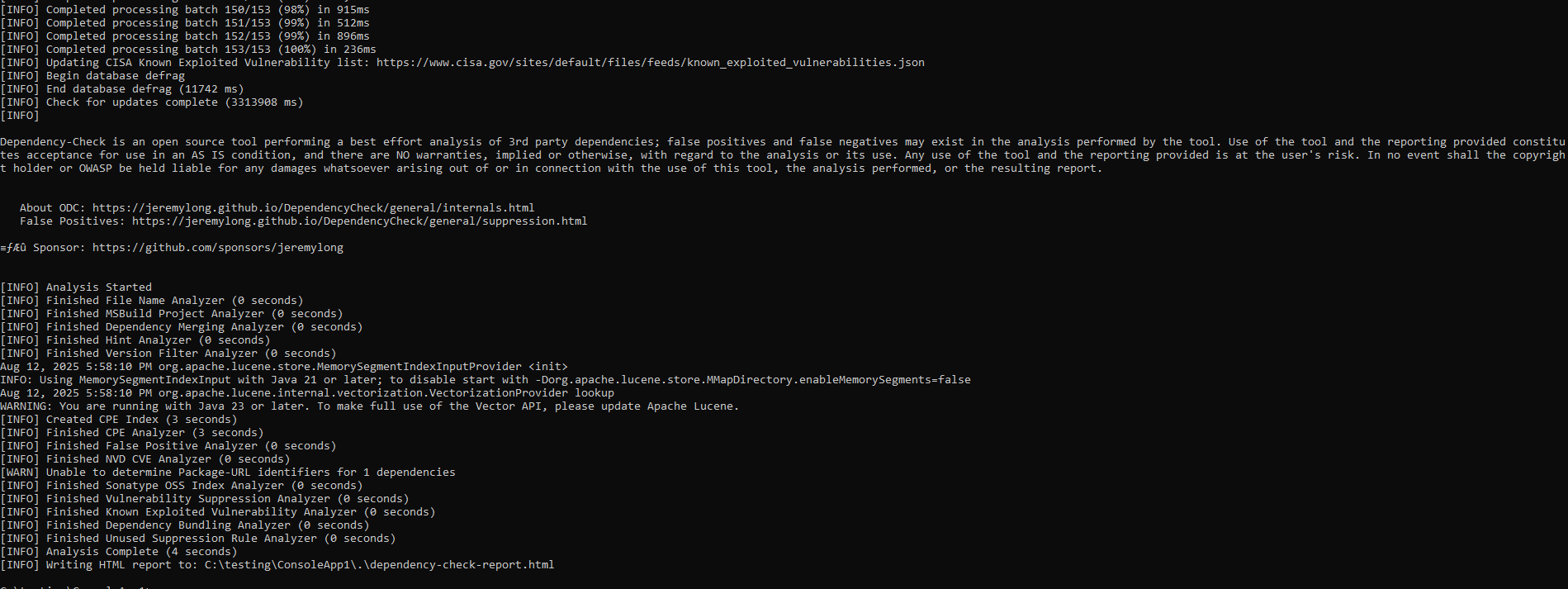


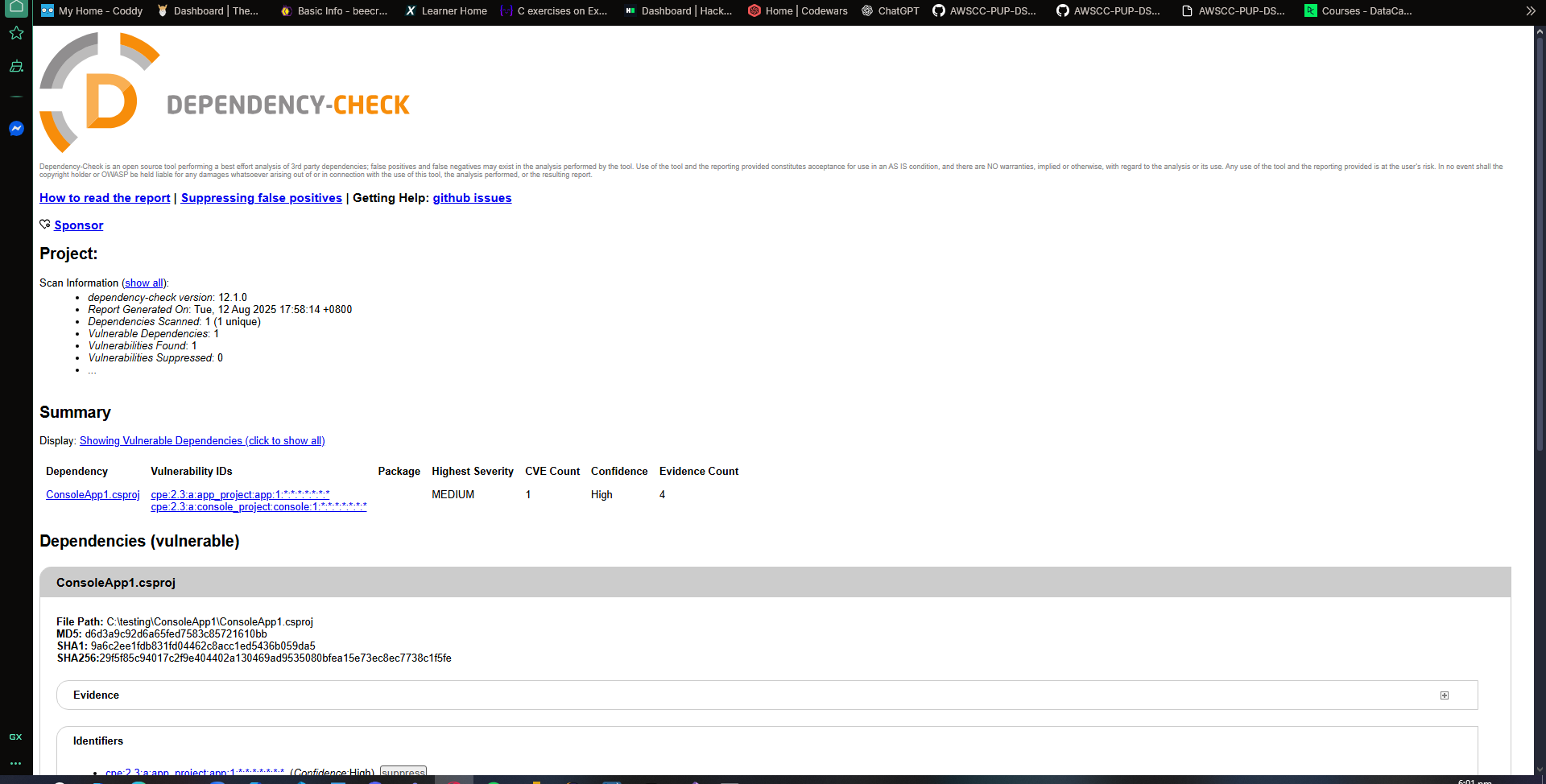
STEP 9

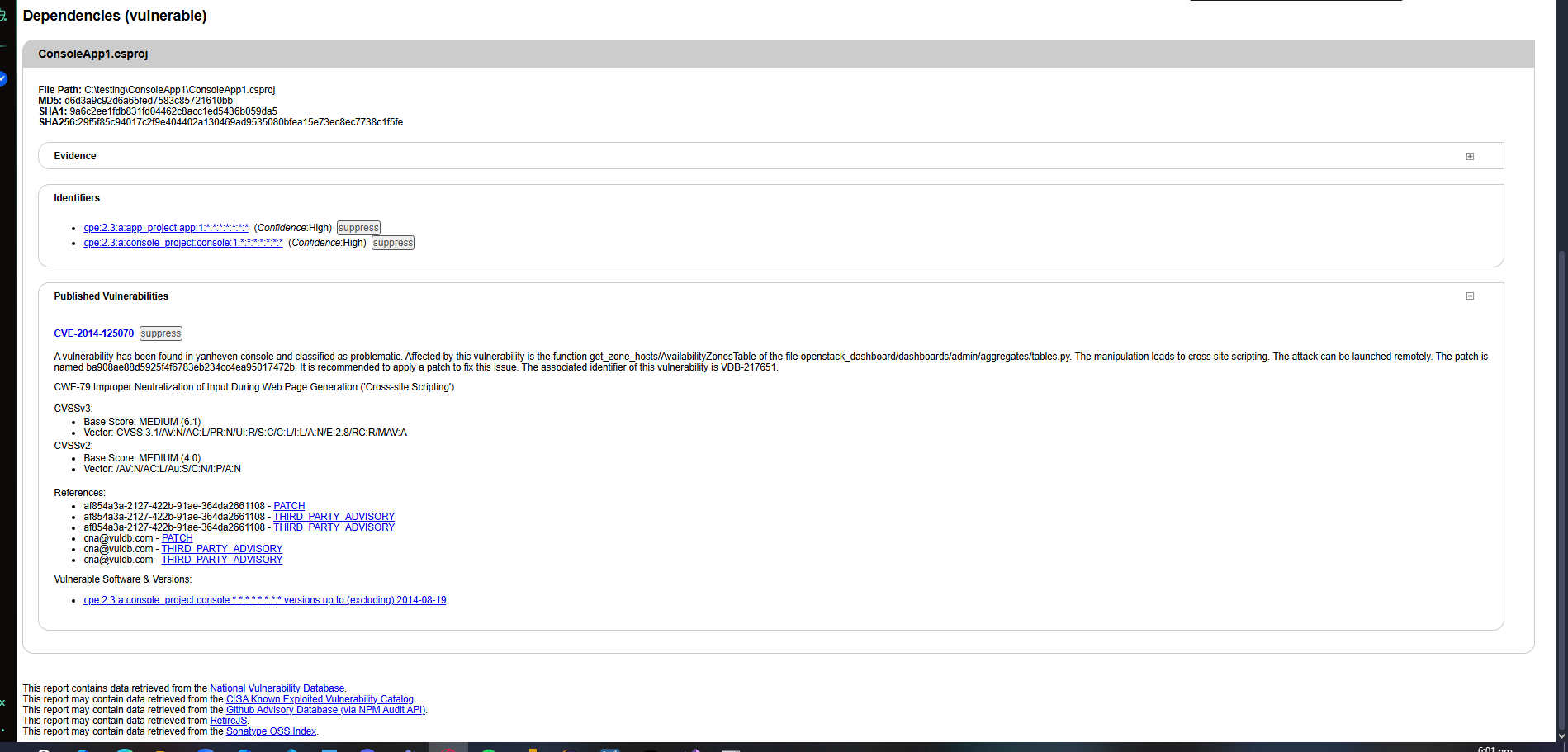


STEP 10









RELFECTION

I executed the Dependency-Check scan from the command line by navigating to my project folder using the cd command and running the dependency-check.bat script located in the C:\dependency-check\bin directory. Initially, I tried running the scan without an NVD API key, but the update process was extremely slow because it was downloading a large number of records with long pauses in between. I decided to stop the process, register my email with the NVD, and obtain an API key to speed up the database update. After some research, I learned how to pass the key as a parameter using the --nvdApiKey flag in the command. Once I restarted the scan with the API key, the download progressed steadily in increments of 10,000 records, which was much faster than before. The scan generated a report in the same directory as my project, listing detected vulnerabilities along with their severity and the files or dependencies where they were found. From reviewing the report, I noticed specific vulnerabilities tied to outdated libraries, which could potentially be addressed by upgrading dependencies to newer, patched versions or replacing them with secure alternatives. One of the main challenges I faced was understanding the update process and why it was slow without a key, but through trial, error, and research, I learned the importance of using the NVD API key for efficiency and the value of keeping the vulnerability database up to date before scanning.