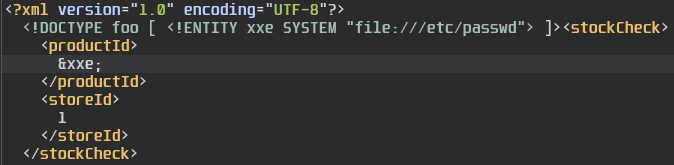
1. Exploiting XXE using external entities to retrieve files:
   * The lab contained a vulnerability in the stock check feature of the application as it parses XML input and returns some response.
   * So, we injected some XML code and modified it to return some system files.
2. Exploiting XXE to perform SSRF attacks:
   * This lab contained a vulnerability in the stock check feature of the application as it parses XML input and returns some response.
   * We will try to retrieve some data by modifying/injecting XML syntax to retrieve useful and confidential information.
   * Which will force the application to do a SSRF attack on the desired URL.
3. Exploiting XInclude to retrieve files:
   * Now in this case the application didn’t control the XML document so we can’t define a DTD to launch a classic XXE attack.
   * We tried exploiting the vulnerability present in the check stock feature by inserting xinclude tag into the product id parameter.
4. Exploiting XXE via image file upload:
   * The lab possessed a vulnerability in the avatar upload functionality of the comment section so we tried to exploit it by uploading a SVG file which is parsed using XML.
   * We uploaded a XML script:

**<?xml version="1.0" standalone="yes"?><!DOCTYPE test [ <!ENTITY xxe SYSTEM "file:///etc/hostname" > ]><svg width="128px" height="128px" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" version="1.1"><text font-size="16" x="0" y="16">&xxe;</text></svg>**

named with .svg extension to the avatar and when we came back to the page we saw the content of the desired file in place of the avatar.