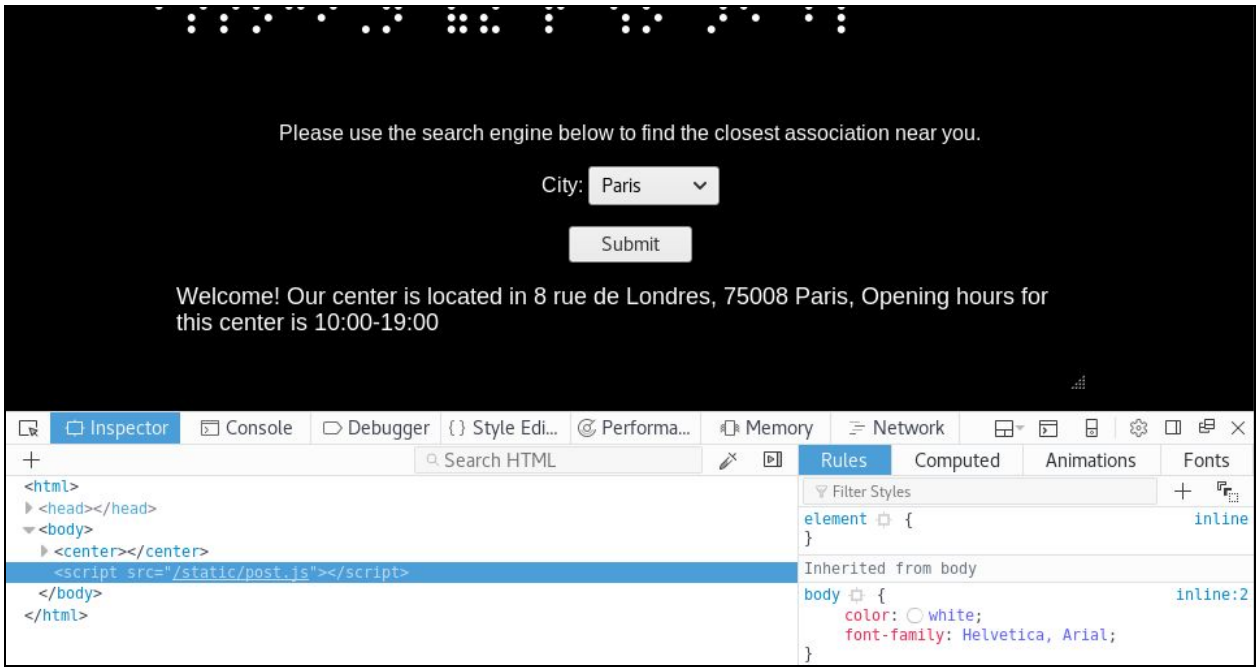
WEB - bnv (155)

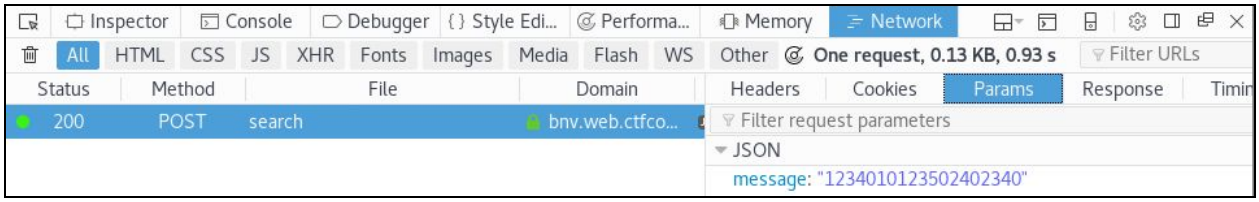
Site: https://bnv.web.ctfcompetition.com/
Topic: Blind XXE Injection
Requirement: BURP Suite

Summary:
1. Inspect the page and click on the (java)script called post.js

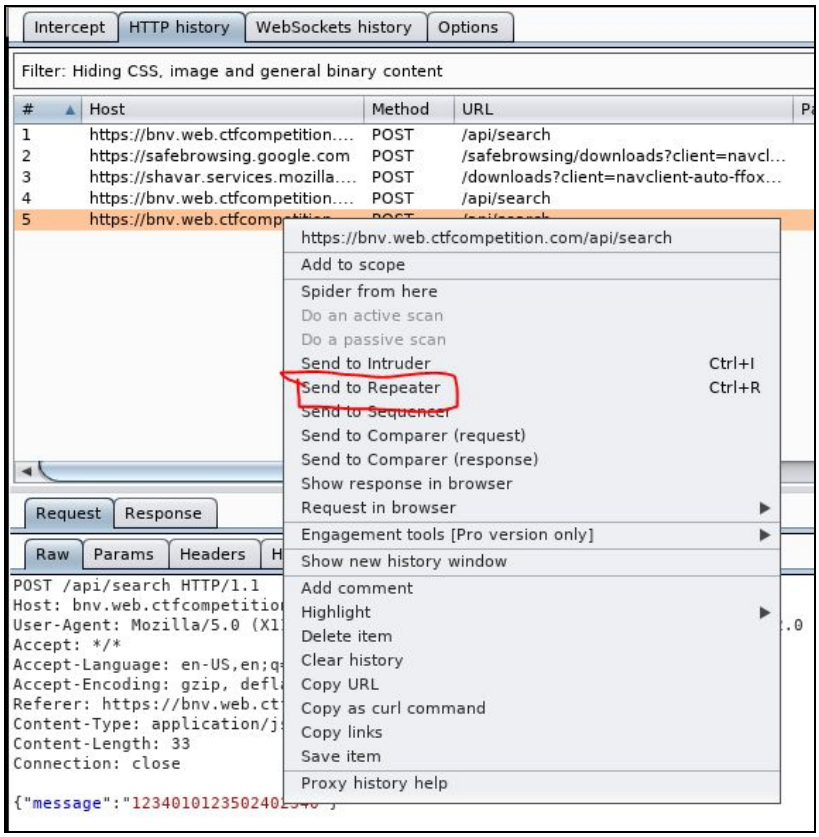


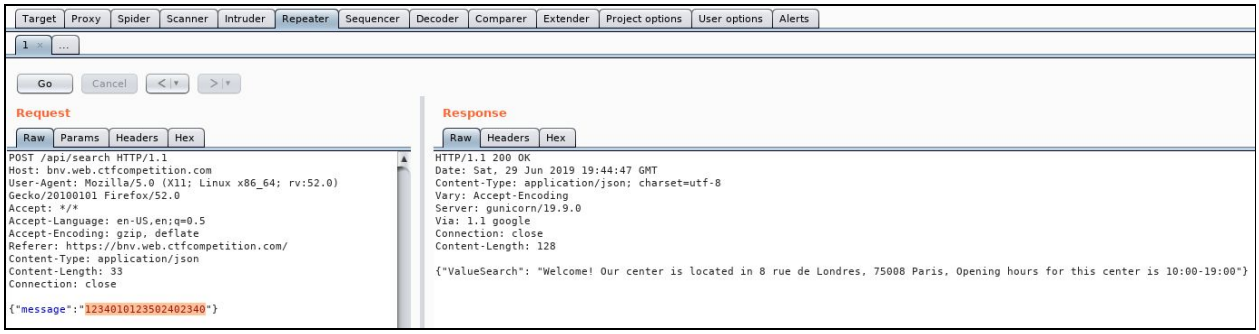
2. The javascript file contains a function call AjaxFormPost(). The array ‘blindvalues’ represents the braille alphabet (https://www.pharmabraille.com).

3. In the Network tab, a POST event will get pop up every time a submit button is hit. Each POST event will contain a JSON message of random numbers. Map the numbers to the braille alphabet. Essentially, the number will vary depending on the location being selected. Example: 135601360123502401401250 == Zurich.



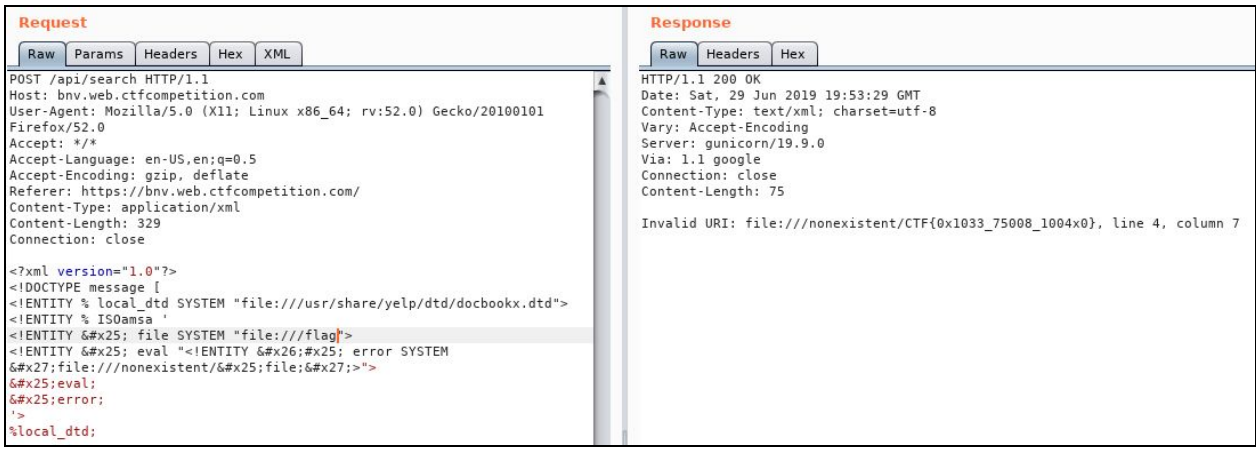
4. A web service with a JSON endpoint may be vulnerable to XML External Entity attacks (XXE), an attack that exploits weakly configured XML parser settings on the server. With the help of BURP Suite, we could send one of the POST events to Repeater and play around with the content.





5. Change the “Content-Type” from ‘application/json’ to ‘application/xml’ and perform any modification to see that the application’s XML data is modifiable. However, the response doesn’t return the values of any defined external entities. Thus, we are dealing with a blind XXE.

6. After a couple of trial-and-error, the proper exploitation technique is by repurposing a local DTD. Essentially, we are trying to trigger an XML parsing error message that contains the contents of the sensitive file by submitting a hybrid DTD (Document Type Definition). However, the flag wasn’t placed in the /etc/passwd file. Thus, we tried to locate everywhere, until finally we found a file called flag.



7. The flag for this challenge is: **CTF{0x1033_75008_1004x0}**

Sources:
https://blog.netspi.com/playing-content-type-xxe-json-endpoints/
https://portswigger.net/web-security/xxe
https://portswigger.net/web-security/xxe/blind
https://portswigger.net/web-security/xxe/blind/lab-xxe-trigger-error-message-by-repurposing-local-dtd