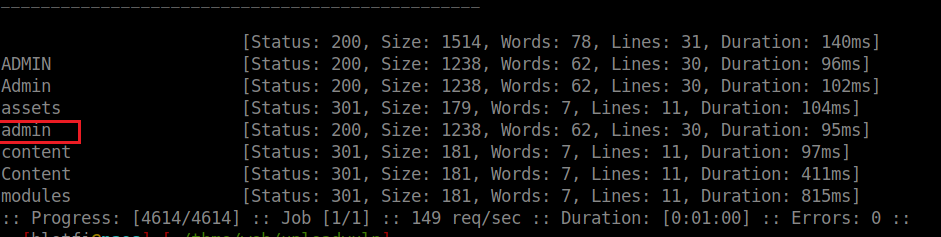
**File Upload Vulns Challenge**

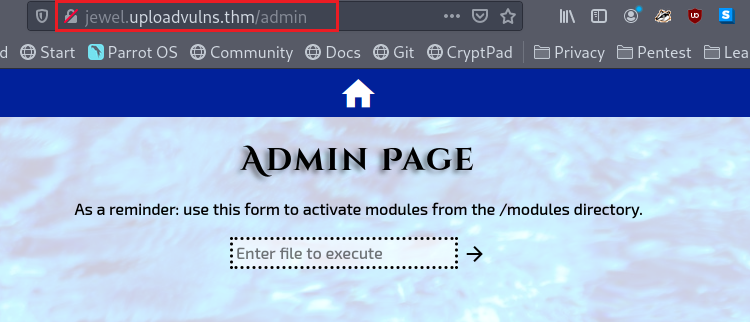
**Enumeration:**

**Fuzzing:**

After fuzzing with ffuf the directories/paths list found is:



* admin: is a path/page used to activate a JS module (file) exist in the server.

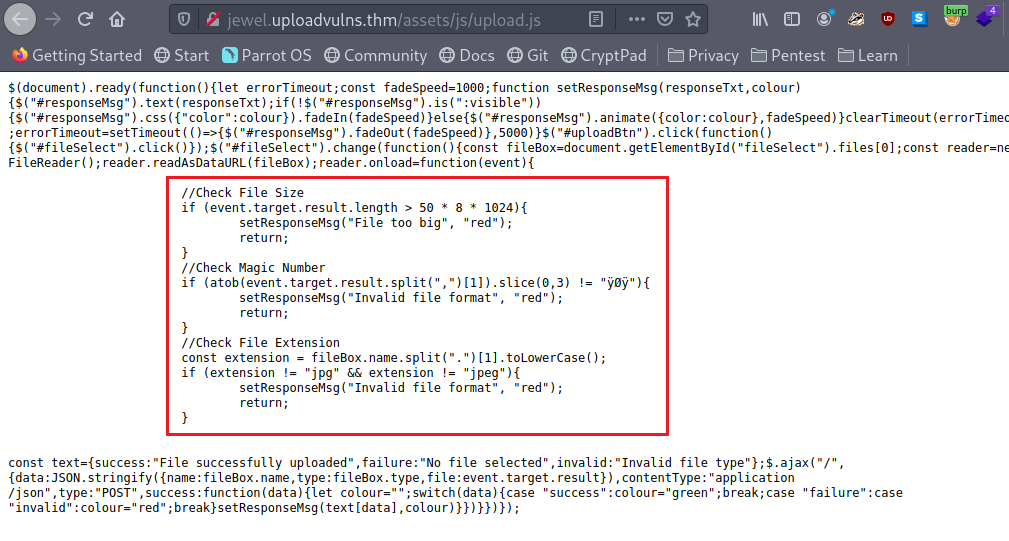


* content is a directory that contains all the uploaded files.

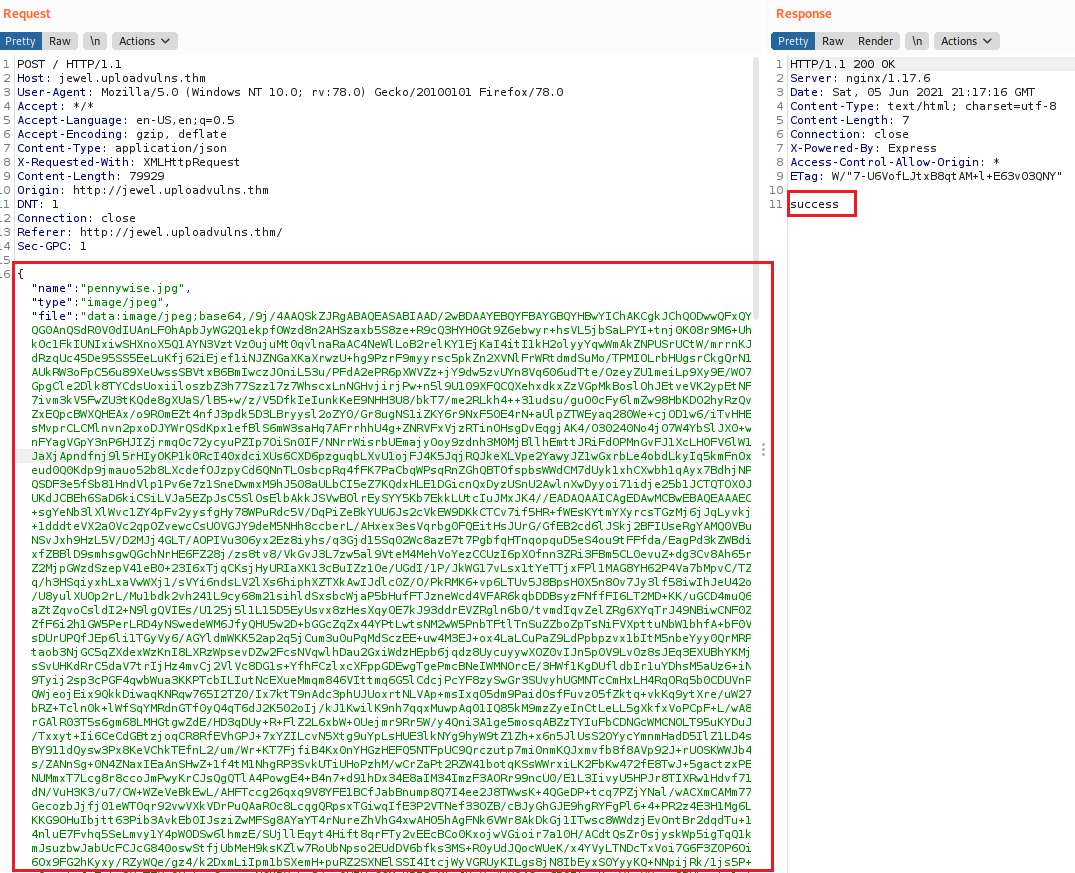
**Test Upload functionality:**

In this section we will understand how file uploading is done.

A client-side file verification is done using the file upload.js, this file verifies the uploaded file’s size, magic number, and extension:



After uploading, the application converts the contents of the file to base64 and sends the encoded data, file name and type to the server in JSON format:



**Exploit:**

To get a web shell we will upload a malicious module and activate it using the admin page, but firstly we need the filename and the path of the uploaded module.

When the server receives the image, it will rename it with one of the words exists in the “UploadVulnsWordlist.txt” wordlist, and move it to the content directory, so to find for the uploaded image we will fuzz the content directory using the help wordlist with gobuster or ffuf.

**Step 1: upload and intercept**

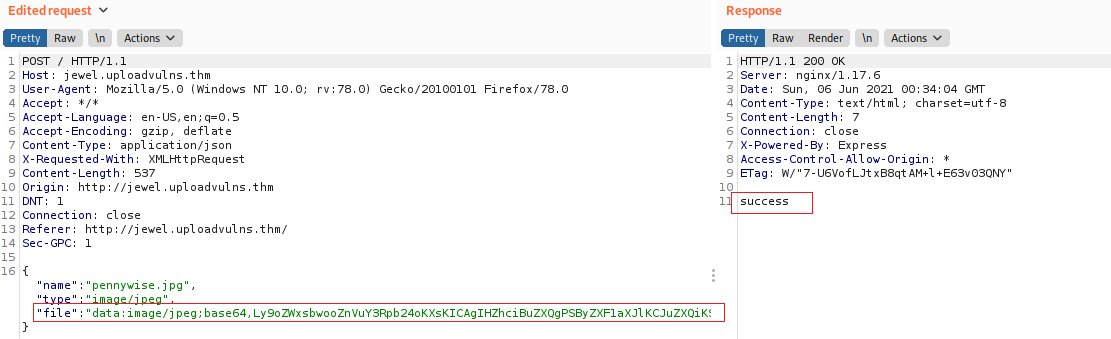
After uploading a valid image, we need to intercept the request to inject our base64 encoded payload.

Our payload is here <https://github.com/appsecco/vulnerable-apps/tree/master/node-reverse-shell>, Note that I added the first line (comment) to bypass a content check.

After adjusting your payload, you need to encode it with base64:

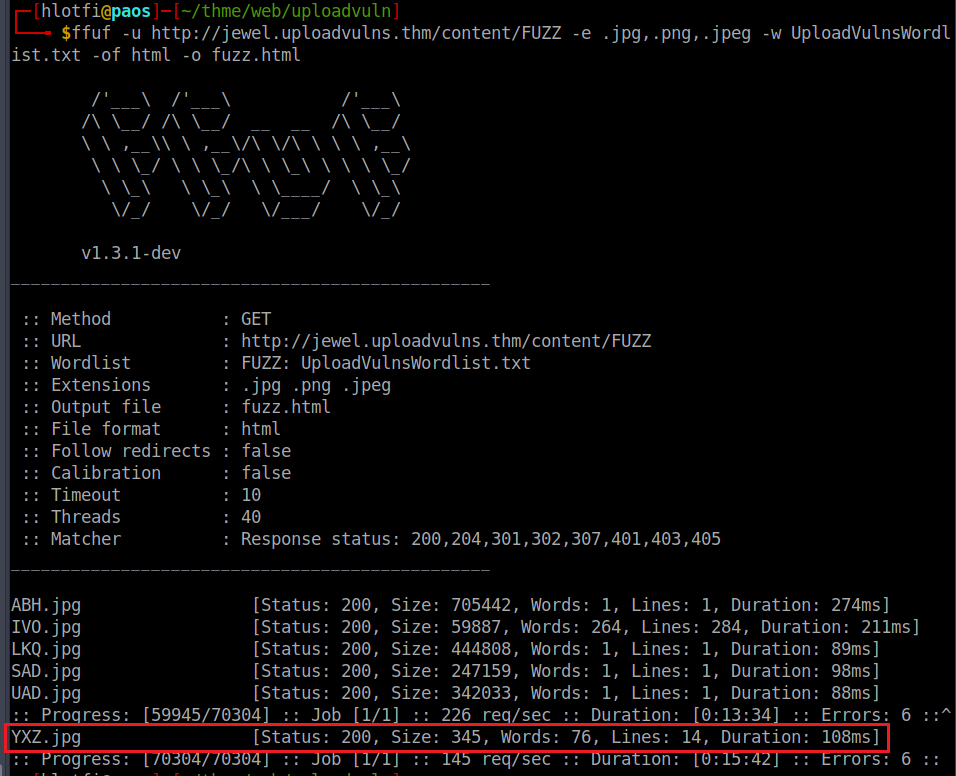


Copy the encoded data and past it to the intercepted request, note that is not necessary to edit the file name and type:

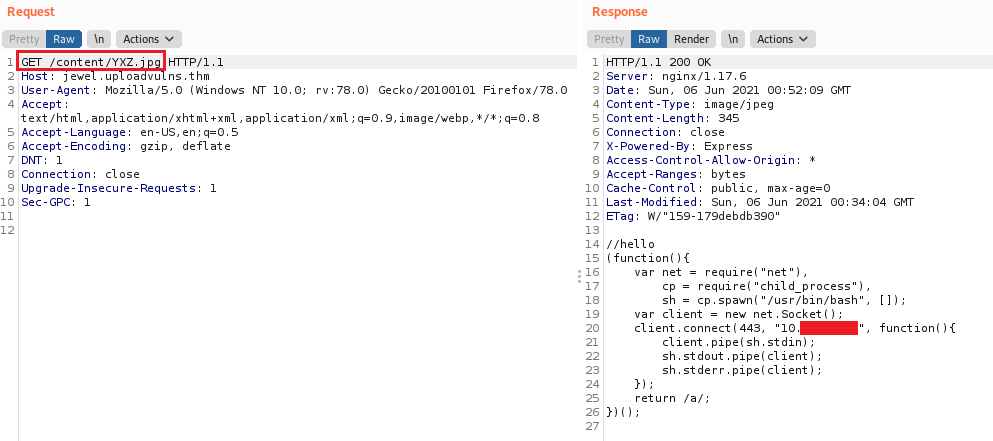


**Step two: search for it**

After uploading our malicious file, we need to search for it:

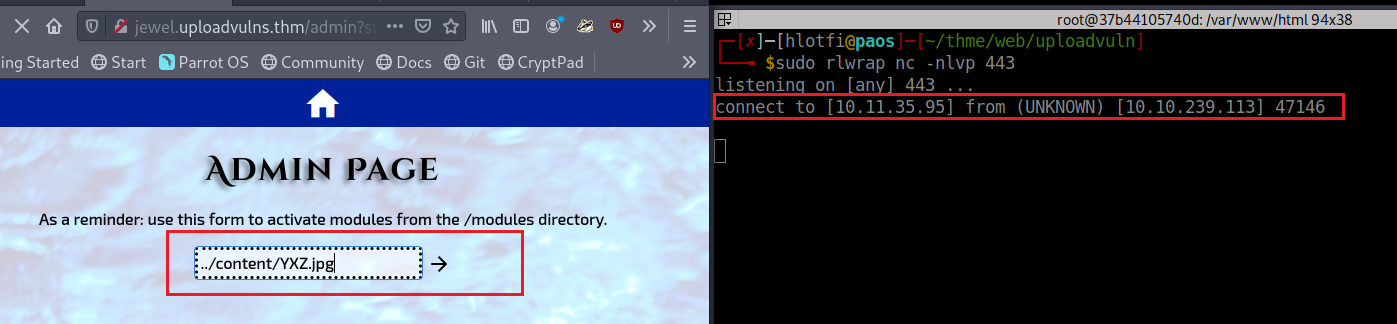


As you can see, our malicious file is renamed to “YXZ.jpg”:



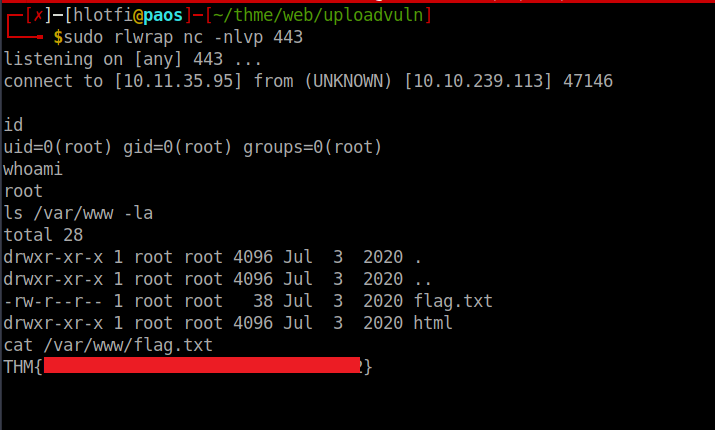
**Step three: activate it**

Now we need to activate this module using the admin page, don’t forget to set your netcat listener:



As you can see, we successfully got a web shell.

Get your flag:



Thanks for reading, sorry this is my first writing hope this helps you 😉.