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( s: "/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
38            }catch (FileUploadException e) {
39                e.printStackTrace();
40            }
41            Iterator iter = formLists.iterator();
42            while (iter.hasNext()){
43                formItem = (FileItem) iter.next();
44                if (!formItem.isFormField()){
45                    if (formItem.getName()!=null&&!formItem.getName().equals("")){
46                        long unloadSize = formItem.getSize();
47                        if (unloadSize>fileSize){
48                            System.out.println("文件大小超过100M");
49                            return;
50                        }
51                        String unloadFileName = formItem.getName();
52                        System.out.println(unloadFileName);
53                        String newFileName = savePath + File.separator + unloadFileName;
54                        File file = new File(newFileName);
55                        try{
56                            formItem.write(file);
```

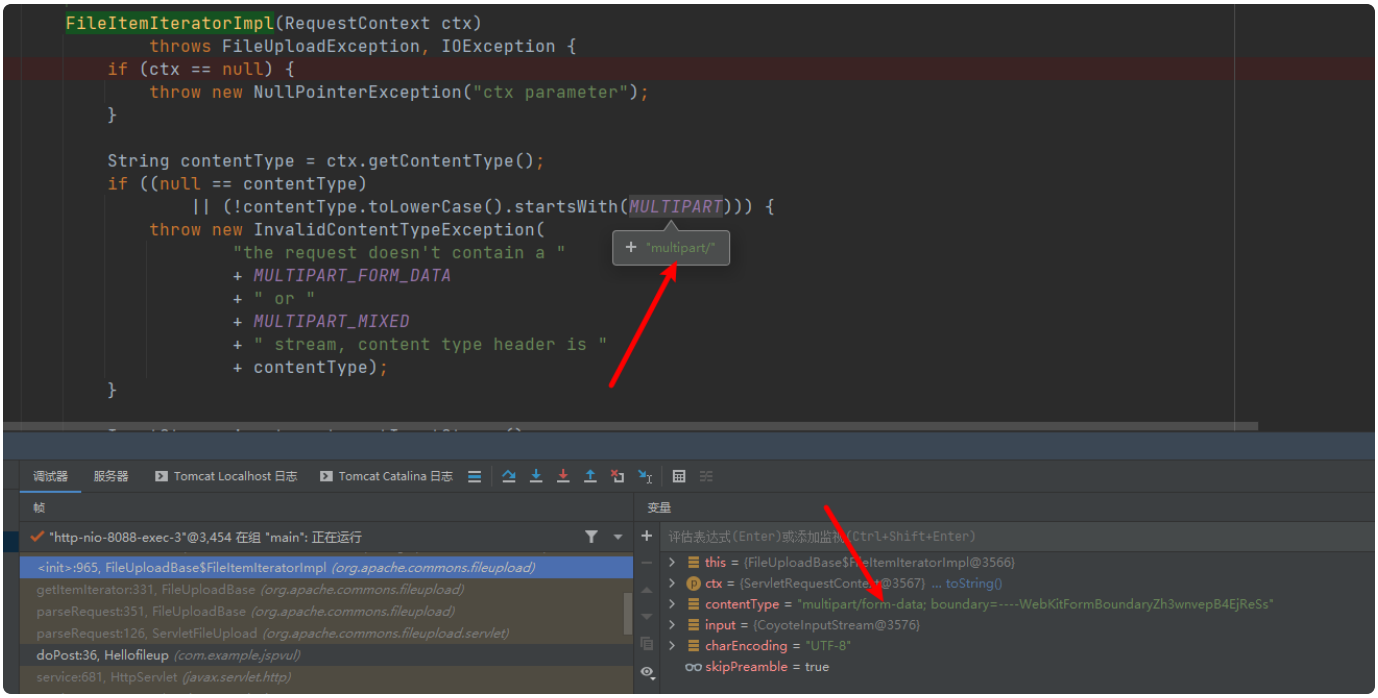
注意到下面肯定是以及解析完了，那么解析的地方肯定在箭头处。

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

```java
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);
            try {
```

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



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#getFileName(java.lang.String)`

```java
private String getFileName(String pContentDisposition) {  pContentDisposition: "form-data; name="file"; filename="11111.jsp""
    String fileName = null;  fileName: null
    if (pContentDisposition != null) {
        String cdl = pContentDisposition.toLowerCase();  cdl: "form-data; name="file"; filename="11111.jsp""
        if (cdl.startsWith(FORM_DATA) || cdl.startsWith(ATTACHMENT)) {  cdl: "form-data; name="file"; filename="11111.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="11111.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`

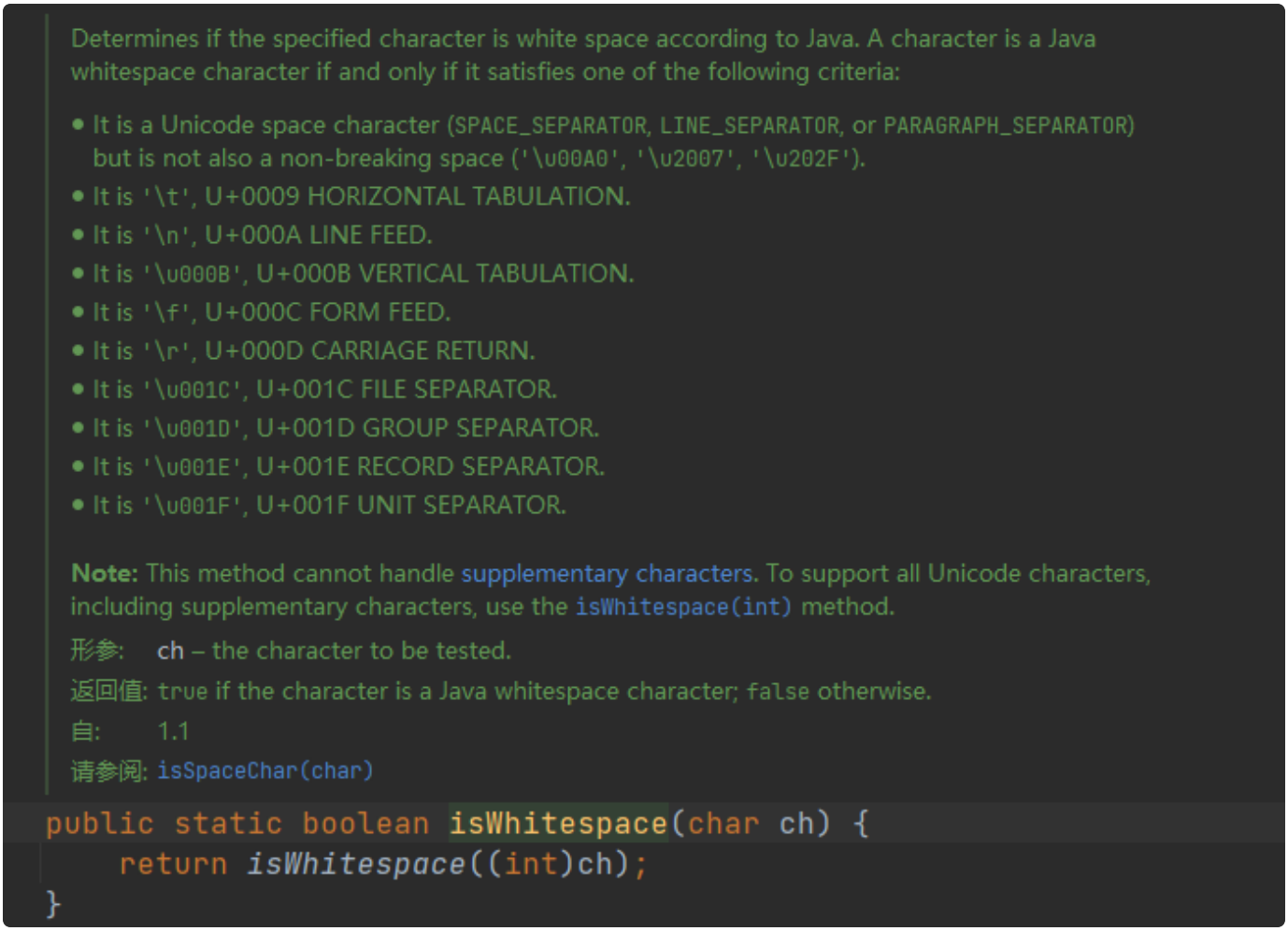
```
private boolean isOneOf(char ch, final char[] charray) {
    boolean result = false;
    for (int i = 0; i < charray.length; i++) {
        if (ch == charray[i]) {
            result = true;
            break;
        }
    }
    return result;
}
```

geToken

```java
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`

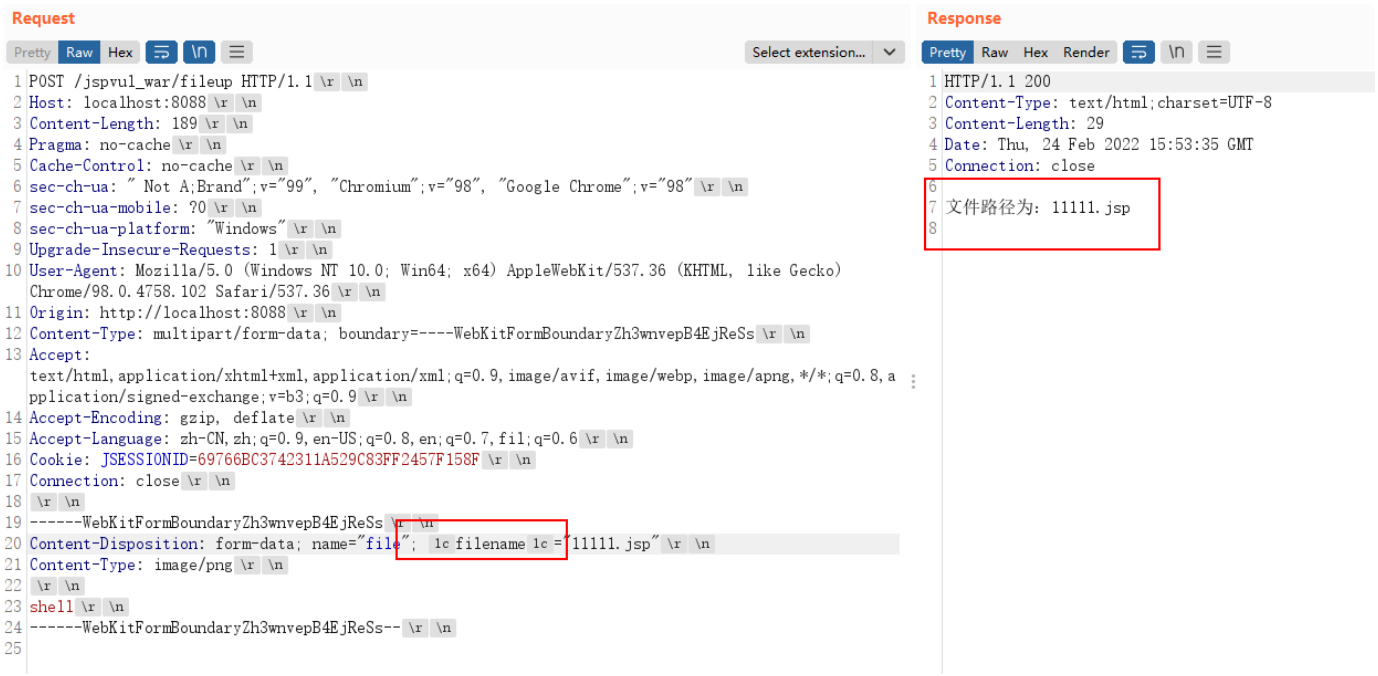


这个是判断师傅是空白字符而且不止我们常用的空格还包括

Plain Text 复制代码

```
%20
%09
%0a
%0b
%0c
%0d
%1c
%1d
%1e
%1f
```

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。



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。