Struts 2 S2-045 Jakarta插件远程代码执行漏洞加固方法

近日，安恒信息安全研究院WEBIN实验室高级安全研究员nike.zheng发现著名J2EE框架-Struts2存在远程代码执行的严重漏洞。

**漏洞编号：**

S2-045，CVE-2017-5638

**漏洞名称：**

基于 Jakarta plugin插件的Struts远程代码执行漏洞

**官方评级：**

高危

**漏洞描述：**

Apache Struts 2被曝出存在远程命令执行漏洞，漏洞编号S2-045，CVE编号CVE-2017-5638，在使用基于Jakarta插件的文件上传功能时，有可能存在远程命令执行，导致系统被黑客入侵。恶意用户可在上传文件时通过修改HTTP请求头中的Content-Type值来触发该漏洞，进而执行系统命令。

**漏洞利用条件和方式：**

黑客通过Jakarta 文件上传插件实现远程利用该漏洞执行代码。

1.基于Jakarta（Jakarta Multipart parser）插件的文件上传功能

2.恶意攻击者精心构造Content-Type的值

**漏洞影响范围：**

Struts 2.3.5 – Struts 2.3.31

Struts 2.5 – Struts 2.5.10

**加固方式如下：**

 通过判断Content-Type头是否为白名单类型，来限制非法Content-Type的攻击。

加固代码：

|  |
| --- |
| **import** java.io.IOException;**import** javax.servlet.Filter;**import** javax.servlet.FilterChain;**import** javax.servlet.FilterConfig;**import** javax.servlet.ServletException;**import** javax.servlet.ServletRequest;**import** javax.servlet.ServletResponse;**import** javax.servlet.http.HttpServlet;**import** javax.servlet.http.HttpServletRequest;**import** javax.servlet.http.HttpServletResponse;**public** **class** SecurityFilter **extends** HttpServlet **implements** Filter { /\*\* \*  \*/ **private** **static** **final** **long** ***serialVersionUID*** = 1L;   **public** **final** String www\_url\_encode= "application/x-www-form-urlencoded"; **public** **final** String mul\_data= "multipart/form-data "; **public** **final** String txt\_pla= "text/plain"; **public** **void** doFilter(ServletRequest arg0, ServletResponse arg1, FilterChain arg2) **throws** IOException, ServletException { HttpServletRequest request = (HttpServletRequest) arg0; HttpServletResponse response = (HttpServletResponse) arg1;  String contenType=request.getHeader("conTent-type");  **if**(contenType!=**null**&&!contenType.equals("")&&!contenType.equalsIgnoreCase(www\_url\_encode)&&!contenType.equalsIgnoreCase(mul\_data)&&!contenType.equalsIgnoreCase(txt\_pla)){  response.setContentType("text/html;charset=UTF-8"); response.getWriter().write("非法请求Content-Type！"); **return**; } arg2.doFilter(request, response); } **public** **void** init(FilterConfig arg0) **throws** ServletException { }} |

1. 将Java编译以后的“SecurityFilter.class”（SecurityFilter.java是源代码文件）复制到应用的WEB-INF/classes目录下。
2. 配置Filter

将下面的代码加入WEB-INF/web.xml文件中。

|  |
| --- |
| <filter> <filter-name>SecurityFilter</filter-name> <filter-class>SecurityFilter</filter-class> </filter><filter-mapping> <filter-name>SecurityFilter</filter-name> <url-pattern>/\*</url-pattern></filter-mapping> |

/\*代表拦截所有请求，进行攻击代码检查，\*.action只检查.action结尾的请求。

示例：



1. 重启应用即可