攻防世界 Misc高手进阶区 3分题 3-1

原创

思源湖的鱼 于 2020-11-27 10:36:21 发布 435 收藏 1
分类专栏： ctf 文章标签： ctf 攻防世界 misc 追踪流量

本文链接：https://blog.csdn.net/weixin_44604541/article/details/110222454
版权

ctf 专栏收录该内容

200 篇文章 23 订阅
订阅专栏

前言

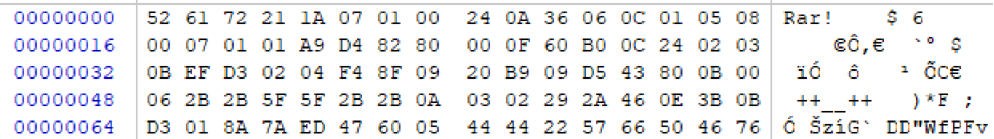
继续ctf的旅程
攻防世界Misc高手进阶区的3分题
本篇是3-1的writeup

发现攻防世界的题目分数是动态的
就仅以做题时的分数为准了

解题过程

下下来一个无后缀文件

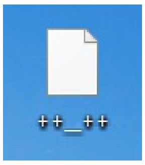
扔进winhex



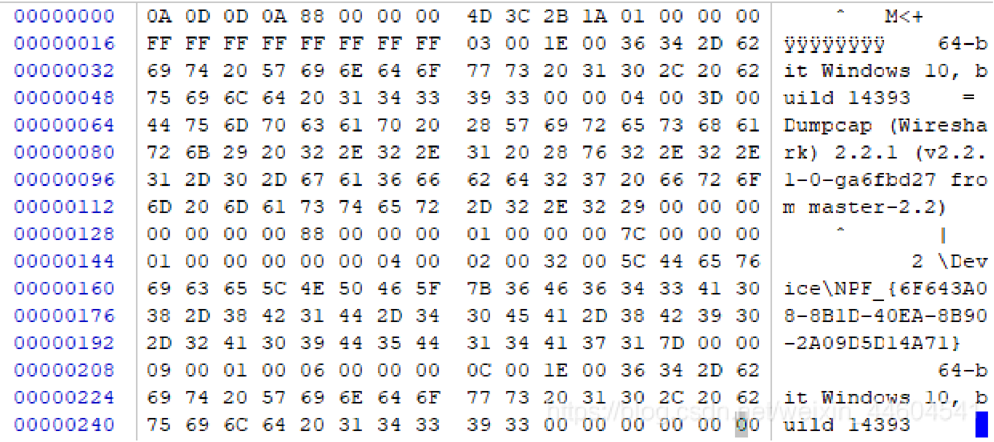
```
00000000  52 61 72 21 1A 07 01 00  24 0A 36 06 0C 01 05 08  Rar!     $ 6
00000016  00 07 01 01 A9 D4 82 80  00 0F 60 B0 0C 24 02 03      ©Ô,€  `° $
00000032  0B EF D3 02 04 F4 8F 09  20 B9 09 D5 43 80 0B 00   ïÓ  ô   ¹ ÕC€
00000048  06 2B 2B 5F 5F 2B 2B 0A  03 02 29 2A 46 0E 3B 0B   ++__++   )*F ;
00000064  D3 01 8A 7A ED 47 60 05  44 44 22 57 66 50 46 76  Ó ŠzíG` DD"WfPFv
```

发现是个rar

改后缀
解压



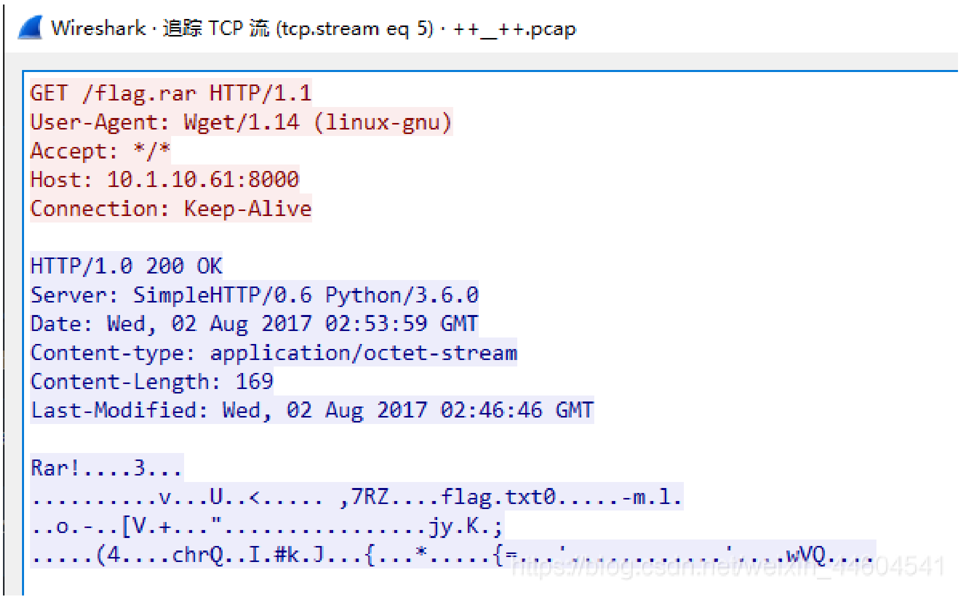
扔进winhex



```
00000000  0A 0D 0D 0A 88 00 00 00  4D 3C 2B 1A 01 00 00 00      ˆ   M<+
00000016  FF FF FF FF FF FF FF FF  03 00 1E 00 36 34 2D 62  ÿÿÿÿÿÿÿÿ    64-b
00000032  69 74 20 57 69 6E 64 6F  77 73 20 31 30 2C 20 62  it Windows 10, b
00000048  75 69 6C 64 20 31 34 33  39 33 00 00 04 00 3D 00  uild 14393    =
00000064  44 75 6D 70 63 61 70 20  28 57 69 72 65 73 68 61  Dumpcap (Wiresha
00000080  72 6B 29 20 32 2E 32 2E  31 20 28 76 32 2E 32 2E  rk) 2.2.1 (v2.2.
00000096  31 2D 30 2D 67 61 36 66  62 64 32 37 20 66 72 6F  1-0-ga6fbd27 fro
00000112  6D 20 6D 61 73 74 65 72  2D 32 2E 32 29 00 00 00  m master-2.2)
00000128  00 00 00 00 88 00 00 00  01 00 00 00 7C 00 00 00      ˆ       |
00000144  01 00 00 00 00 00 04 00  02 00 32 00 5C 44 65 76           2 \Dev
00000160  69 63 65 5C 4E 50 46 5F  7B 36 46 36 34 33 41 30  ice\NPF_{6F643A0
00000176  38 2D 38 42 31 44 2D 34  30 45 41 2D 38 42 39 30  8-8B1D-40EA-8B90
00000192  2D 32 41 30 39 44 35 44  31 34 41 37 31 7D 00 00  -2A09D5D14A71}
00000208  09 00 01 00 06 00 00 00  0C 00 1E 00 36 34 2D 62              64-b
00000224  69 74 20 57 69 6E 64 6F  77 73 20 31 30 2C 20 62  it Windows 10, b
00000240  75 69 6C 64 20 31 34 33  39 33 00 00 00 00 00 00  uild 14393
```

发现是个流量包

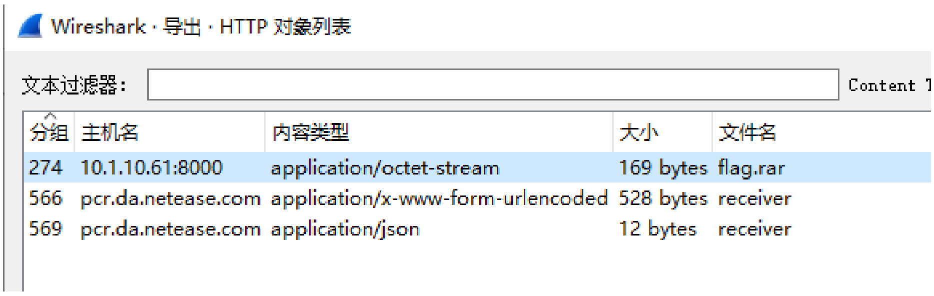
在第5个TCP流中发现flag.rar



Wireshark · 追踪 TCP 流 (tcp.stream eq 5) · ++_++.pcap

```
GET /flag.rar HTTP/1.1
User-Agent: Wget/1.14 (linux-gnu)
Accept: */*
Host: 10.1.10.61:8000
Connection: Keep-Alive

HTTP/1.0 200 OK
Server: SimpleHTTP/0.6 Python/3.6.0
Date: Wed, 02 Aug 2017 02:53:59 GMT
Content-type: application/octet-stream
Content-Length: 169
Last-Modified: Wed, 02 Aug 2017 02:46:46 GMT

Rar!....3...
..........v...U..<..... ,7RZ....flag.txt0.....-m.l.
..o.-..[V.+..."................jy.K.;
.....(4....chrQ..I.#k.J...{...*.....{=...'...........'....wVQ....
```

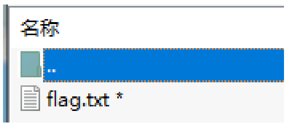
导出



Wireshark · 导出 · HTTP 对象列表

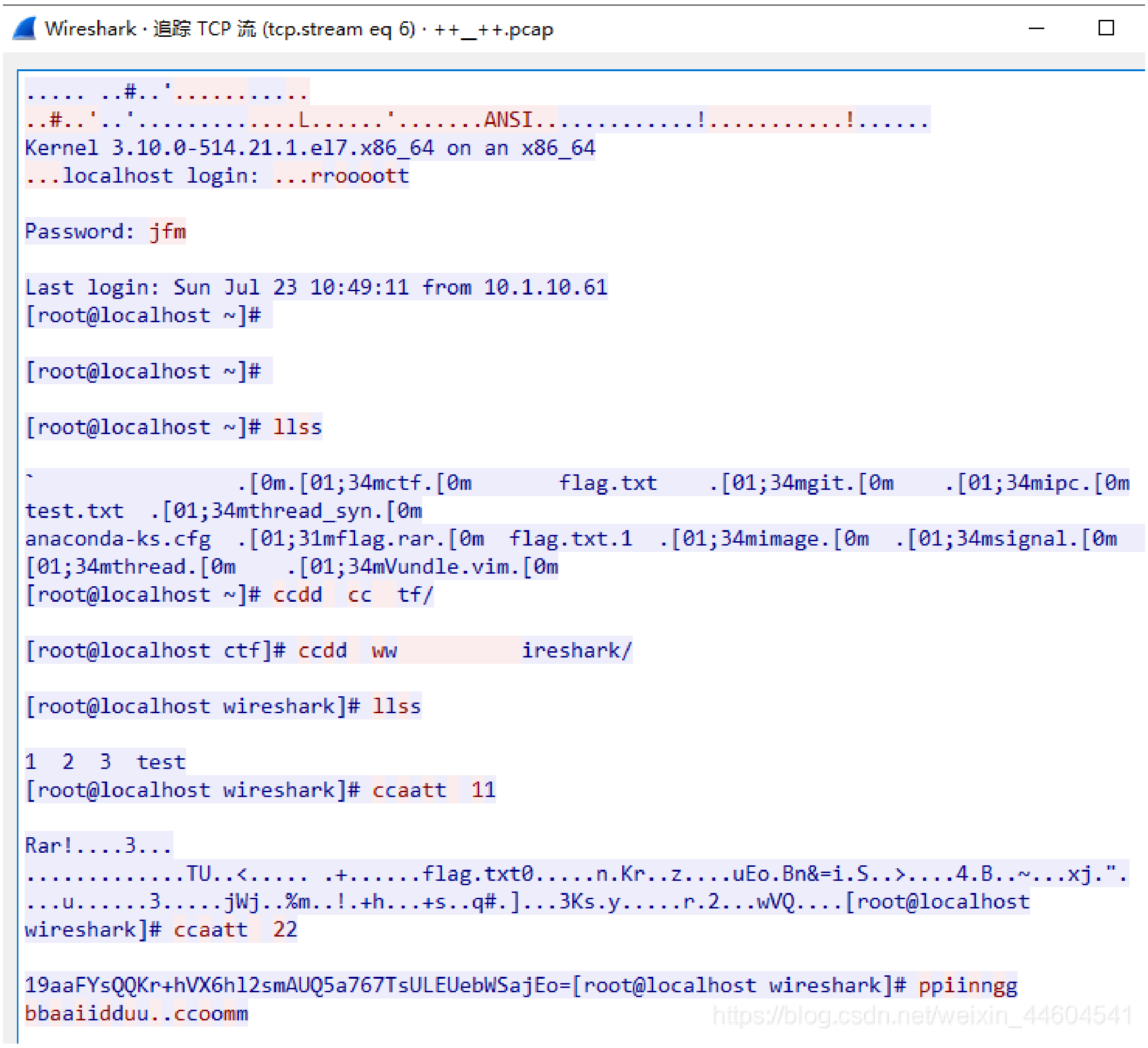
文本过滤器： Content T

分组	主机名	内容类型	大小	文件名
274	10.1.10.61:8000	application/octet-stream	169 bytes	flag.rar
566	pcr.da.netease.com	application/x-www-form-urlencoded	528 bytes	receiver
569	pcr.da.netease.com	application/json	12 bytes	receiver



解压要密码
看了看不是伪加密
那密码还在流量包里

在第6个TCP流中发现一些linux指令
一个base64
一段python



```
..... ..#..'..........
..#..'..'..............L......'.......ANSI............!...........!......
Kernel 3.10.0-514.21.1.el7.x86_64 on an x86_64
...localhost login: ...rroooott

Password: jfm

Last login: Sun Jul 23 10:49:11 from 10.1.10.61
[root@localhost ~]#

[root@localhost ~]#

[root@localhost ~]# llss

`                 .[0m.[01;34mctf.[0m       flag.txt    .[01;34mgit.[0m    .[01;34mipc.[0m
test.txt  .[01;34mthread_syn.[0m
anaconda-ks.cfg  .[01;31mflag.rar.[0m  flag.txt.1  .[01;34mimage.[0m  .[01;34msignal.[0m
[01;34mthread.[0m    .[01;34mVundle.vim.[0m
[root@localhost ~]# ccdd  cc  tf/

[root@localhost ctf]# ccdd  ww          ireshark/

[root@localhost wireshark]# llss

1  2  3  test
[root@localhost wireshark]# ccaatt  11

Rar!....3...
.............TU..<..... .+......flag.txt0.....n.Kr..z....uEo.Bn&=i.S..>....4.B..~...xj.".
...u......3.....jWj..%m..!.+h...+s..q#.]...3Ks.y.....r.2...wVQ....[root@localhost
wireshark]# ccaatt  22

19aaFYsQQKr+hVX6hl2smAUQ5a767TsULEUebWSajEo=[root@localhost wireshark]# ppiinngg
bbaaiidduu..ccoomm
```

整理一下

base64：`19aaFYsQQKr+hVX6hl2smAUQ5a767TsULEUebWSajEo=`

python段

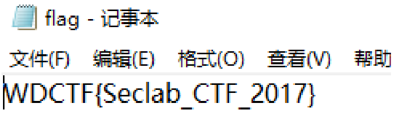
```
# coding:utf-8

__author__ = 'YFP'

from Crypto import Random
from Crypto.Cipher import AES
import sys
import base64

IV = 'QWERTYUIOPASDFGH'
def decrypt(encrypted):
    aes = AES.new(IV, AES.MODE_CBC, IV)
    return aes.decrypt(encrypted)

def encrypt(message):
    length = 16
    count = len(message)
    padding = length - (count % length)
    message = message + '\0' * padding
    aes = AES.new(IV, AES.MODE_CBC, IV)
    return aes.encrypt(message)

str = 'this is a test'
example = encrypt(str)
print(decrypt(example))
```

是个AES加密
把base64用上

```
# coding:utf-8

__author__ = 'YFP'

from Crypto import Random
from Crypto.Cipher import AES
import sys
import base64

IV = 'QWERTYUIOPASDFGH'
def decrypt(encrypted):
    aes = AES.new(IV, AES.MODE_CBC, IV)
    return aes.decrypt(encrypted)

def encrypt(message):
    length = 16
    count = len(message)
    padding = length - (count % length)
    message = message + '\0' * padding
    aes = AES.new(IV, AES.MODE_CBC, IV)
    return aes.encrypt(message)

str = 'this is a test'
example = encrypt(str)
print(decrypt(example))
s='19aaFYsQQKr+hVX6hl2smAUQ5a767TsULEUebWSajEo='
flag=base64.b64decode(s)
print(decrypt(flag))
```

得到密码`passwd={No_One_Can_Decrypt_Me}`

解压



WDCTF{Seclab_CTF_2017}

得到flag

结语

简单追踪流量