攻防世界misc高手区部分writeup



title: 攻防世界misc高手区部分writeup tags: ctf categories: ctf

1.easycap

下载下来是一个流量包,用wireshark打开,搜索关键字flag,第二行追踪tcp流,

No.	Time		Source	Destination	Protocol		Leng	th Ir	nfo			
	1 0.000000		172.31.98.199	192.155.81.86	ТСР			74 4	16046→7890	[SYN]	Seq=0	Win=29
	2 0.029197		192.155.81.86	172.31.98.199	тср			74 7	7890→46046	[SYN,	ACK]	Seq=0 A
	3 0.029275	Follow TCP Stream (tcp.stream.eg.0)			_		×	5046→7890	[ACK]	Seq=1	Ack=1
	4 22.722541							~	5046→7890	[PSH,	ACK]	Seq=1 A
	5 22.749416	Stream Content							890→46046	[ACK]	Seq=1	Ack=2
	6 23.723048	CLAC:285b87afc86	71 doo07550200d16280	751					5046→7890	[PSH,	ACK]	Seq=2 A
	7 23.753912	PEAG. 363067 at C60	/10220010880	/1					890→46046	[ACK]	Seq=1	Ack=3
	8 24.723642								5046→7890	[PSH,	ACK]	Seq=3 A
	9 24.753844								890→46046	[ACK]	Seq=1	Ack=4
	10 25.724349								5046→7890	[PSH,	ACK]	Seq=4 A
	11 25.753234								890→46046	[ACK]	Seq=1	Ack=5
	12 26.724839								5046→7890	[PSH,	ACK]	Seq=5 A
	13 26.755643								890→46046	[ACK]	Seq=1	Ack=6
	14 27.725043								5046→7890	[PSH,	ACK]	Seq=6 A
	15 27.755928								890→46046	[ACK]	Seq=1	Ack=7
	16 28.725317								5046→7890	[PSH,	ACK]	Seq=7 A
	17 28.756580								890→46046	[ACK]	Seq=1	Ack=8
	10 30 735503								EA16 7000	Encu	ACV1	Coo 0 4

2.Avatar

下载下来是一张图片,这里考察outguess算法,在kali下载outguess, apt-get install outguess,

"We should blow up the bridge at midnight"即为flag。

```
root@kali:~/Downloads/misc higher# outguess -r b985d98d87b24ff1b17fc02ffc446b2c.jpg out
Reading b985d98d87b24ff1b17fc02ffc446b2c.jpg....
Extracting usable bits: 28734 bits
Steg retrieve: seed: 94, len: 41
root@kali:~/Downloads/misc higher# cat out
We should blow up the bridge at midnight
```

3.What-is-this

下载个解压包,解压后是两张图片,用stegsolve打开其中一张,用image combiner进行xor运算,AZADI TOWER即为flag



4.Get-the-key.txt

下载来一个解压包,解压后的文件不知道是什么,用file命令查看一下,是磁盘文件,那就先挂载下

root@kali:~/Downloads/misc higher/4# file forensic100
forensic100: Linux rev 1.0 ext2 filesystem data, UUID=0b92a753-7ec9-4b20-8c0b-79c1fa140869
root@kali:~/Downloads/misc higher/4# mount forensic100 /mnt/
root@kali:~/Downloads/misc higher/4# []

下载后有一大堆乱七八糟,用grep命令查找关键字"key.txt",在1文件,cat查看是乱码,file查看是什么文件,是 压缩文件,gunzip查看flag

GO GG	@Y@fi	root@	kali:	/mnt#	grep) - r k	ey.tx [.]	t										
匹配	到二进	制文	件 1															
root	@kali	:/mnt	# fil	e 1														
1: g	zip c	ompre	ssed	data,	was	"key.	txt",	last	modi	fied:	Wed	Oct :	1 06	:00:	52 20	014,	fro	n Unix, original
size	30																	-
root	@kali	:/mnt	# gun	zip <	1													
SECO	:0N{@]	NL7n+	-s75F	rET]v	U=7Z}													
root	@kali	:/mnt	#ls															
1	110	122	134	146	158	17	181	193	204	216	228	24	31	43	55	67	79	90
10	111	123	135	147	159	170	182	194	205	217	229	240	32	44	56	68	8	91
100	⊒112 立	124	136	148	16	171	183	195	206	218	23	241	33	45	57	69	80	92
101	113	125	137	149	160	172	184	196	207	219	230	242	34	46	58	7	81	93
102	114	126	138	15	161	173	185	197	208	22	231	243	35	47	59	70	82	94
103	115	127	139	150	162	174	186	198	209	220	232	244	36	48	6	71	83	95
104	116	128	14	151	163	175	187	199	21	221	233	25	37	49	60	72	84	96
105	117	129	140	152	164	176	188	2	210	222	234	26	38	5	61	73	85	97
106	118	13	141	153	165	177	189	20	211	223	235	27	39	50	62	74	86	98
107	119	130	142	154	166	178	19	200	212	224	236	28	4	51	63	75	87	99
108	12	131	143	155	167	179	190	201	213	225	237	29	40	52	64	76	88	lost+found
109	120	132	144	156	168	18	191	202	214	226	238	3	41	53	65	77	89	
11	121	133	145	157	169	180	192	203	215	227	230	30	42	54	66	78	٩	

将Z2dRQGdRMWZxaDBvaHRqcHRfc3d7Z2ZoZ3MjfQ==**base64转码** 为ggQ@gQ1fqh0ohtjpt_sw{gfhgs#},是凯撒+栅栏,解题关键是flag格式ssctf{},所以先凯撒得到 ssC@sC1rct0atfvbf_ei{srtse#}栅栏得到ssctf{ssCtf_seC10ver#@rabit}

6.Training-Stegano-1

下载是一张图片,记事本打开,文本显示"passwd:steganol",flag即为steganol

7.Test-flag-please-ignore

将666cc61677b68656c6c6f5f776f726c647d转换为16进制字符串flag{hello_world}

8.Excaliflag

下载图片后,用stegSolve打开图片

🛓 StegSolve 1.3 by Caesum			_	×
File Analyse Help				
Gray bits				
SORAD	MM - HA			

9.glance-50

下载来是一张gif图片,分解gif,使用convert命令分解,convert 33e3d14fb67a44f4ad1378149fff1d9a.gif flag.png共分解出200张图片

	2 th	—————————————————————————————————————	
		• \\\	
	🙀 flag-0.png	1.6 Ki	В
	flag-1.png	1.7 KE	В
	flag-2.png	1.7 KE	В
	flag-3.png	1.6 KI	В
	Flag-4.png	1.7 KE	в
	Flag-5.png	1.6 Ki	В
	Fig-6.png	1.7 KE	в
	flag-7.png	1.7 KB	в
	Fig-8.png	1.6 Ki	В
	Flag-9.png	1.7 KE	В
	Flag-10.png	1.7 KE	В
	Flag-11.png	1.7 KE	в
	Flag-12.png	1.7 KE	В
	Flag-13.png	1.7 KB	В
	Flag-14.png	1.7 KB	В
	Rig-15.png	1.7 KE	в
	Rag-16.png	1.7 KE	в
23			- F -

连接这些图片,使用montage命令, montage flag*.png -tile x1 -geometry +0+0 flag.png -tile是拼接时每行和每列的图片数,这里用x1,就是只一行

-geometry是首选每个图和边框尺寸,我们边框为0,图照原始尺寸即可



*这里参考了: https://blog.csdn.net/zz_Caleb/article/details/89490494

10.4-2

下载后是一个txt文件,里面是不知道是什么,到这个网站https://quipqiup.com/进行词频分析,flag即为flag{classical-cipher_is_not_security_hs}

quipqiup

quipqiup is a fast and automated cryptogram solver by Edwin Olson. It can solve simple substitution ciphers often found in newspapers, including puzzles like cryptoquips (in which word boundaries are preserved) and patristocrats (inwhi chwor dboun darie saren t).

Puzz	le:	
Eg q Ew 1	nlyjtenzydl tm fgtk jds	. z umaujejmjetg qeydsn eu z bsjdtx tw sgqtxegc al kdeqd mgeju tw yrzegjsoj zns nsyrząsx kejd qeydsnjsoj s kzl tw sgqtxegc m kerr csj jds wrzc kdeqd eu qrzuueqzr-qeydsn_eu_gtj_usqmnejl_du
Clue	S: For example	GER OVWETHF
		auto
		Calva
		Solve
0	-1.593	In cryptography a substitution cipher is a ?ethod of encoding by which units of plaintext are replaced with ciphertext If you know the way of encoding u will get the flag which is classical-cipher_is_not_security_hs
1	-1.785	In cryptography a substitution cipher is a ?ethod of encoding by which units of plaintext are replaced with ciphertext If you ?now the way of encoding u will get the flag which is classical-cipher_is_not_security_hs
2	-1.889	In cryptography a sumstitution cipher is a ?ethod of encoding my which units of plaintext are replaced with ciphertext If you ?now the way of encoding u will get the flag which is classical-cipher_is_not_security_hs

3 -1.944 In cryptography a substitution cipher is a ?etho? of enco?ing by which units of plaintext are replace? with ciphertext If you ?now the way of enco?ing u will get the flag which is classical-cipher_is_not_security_hs

11.misc1

misc1	最佳Writeup由admin提供	📕 WP	● 建议
难度系数:	 ★ 1.0 		
题目来源:	DDCTF-2018		
题目描述: b5e6fd	d4e8e1f4a0f7e1f3a0e6e1f3f4a1a0d4e8e5a0e6ece1e7a0e9f3baa0c4c4c3d4c6fbb9e1e6b3e3b9e4b3b7b7e2b6b1e4b2b6b9e2b1b	1b3b3b7e6b3	b3b0e3b9b3
题目场景:	暂无		
题目附件:	暂无		

猜测16进制转换字符串,直接转失败转不出来,flag为DDCTF{9af3c9d377b61d269b11337f330c935f}

string="d4e8e1f4a0f7e1f3a0e6e1f3f4a1a0d4e8e5a0e6ece1e7a0e9f3baa0c4c4c3d4c6fbb9e1e6b3e3b9e4b3b7b7e2b6b1e4b2b
flag=""
for i in range (0,len(string),2):
 s = "0x" + string[i] + string[i+1]
 flag += chr(int(s,16) -128)
print(flag)

12.eczmbarrass

下载压缩包解压后是一个流量包,用wireshark打开,搜索关键词flag后,追踪tcp流,在里面搜索flag,可以看到 flag

Follow TCP Stream (tcp.stream eq 35) — O
Stream Content
8b.`p.T&.4.>%t F.@.ly.H>B.OY2aUz.jCZ.U2iQ3(&p^.6.D<`.N.T.C5.0\$`.I.SJ!.K.A. <zel.s\. &.<miqc% OmL.p}l.9.h!</zel.s\. &.<miqc%
oh+`.0.
0P
do
[141

13.肥宅快乐题

下载后是一个swf文件,是一个游戏,用potplayer打开定位57帧,出现对话里面有 U1IDe0YzaVpoYWlfa3U0aWxlX1QxMTF9,base64解码后为SYC{F3iZhai_ku4ile_T111},即为flag



14.小小的PDF

下载后是pdf文件,用binwalk分析,有点东西,用foremost分解东西,flag藏在分解的图片里

root@kali:~/D	ownloads/misc	higher/14# binwalk 7e5ab2e7587d4a4abf9c705dfb935a92.pdf
DECIMAL	HEXADECIMAL	DESCRIPTION
0 ∩ ± = = 452 73254 m 81606 82150 cuments 104469 105134 root@kali:~/D Processing: 7 * root@kali:~/D	0x0 0x1C4 0x11E26 0x13EC6 0x140E6 0x19815 0x198AE 0x19AAE 0wnloads/misc 0x00000000000000000000000000000000000	<pre>000 PDF3document, version: "1.4" JPEG image data, JFIF standard 1.01 JPEG image data, JFIF standard 1.01 2lib compressed data, default compression JPEG image data, JFIF standard 1.01 2lib compressed data, default compression Zlib compressed data, default compression tib compressed data, default compression higher/14# foremost 7e5ab2e7587d4a4abf9c705dfb935a92.pdf higher/14#</pre>

									-
			企 主文件夹	Downloads	misc higher	14 output	jpg 👻		
名称									
DPG	00000000.jpg								
DPG	00000143.jpg								
JPG	00000160.jpg	55	% 🗸 🖬		0	0000160.jpg			± 6 图 B
		S	YC{so_so_so	easy}				文件夹 <u>ipg</u> 文件夹 <u>ipg</u> 光圈 曝光 焦距 ISO 测光 相机	
								日期 时间	

15.Cephalopod

下载后是一个流量包,用wireshark打开,flag关键字找到flag.png,但是却弄不出来图片。用binwalk看一下有点 东西,foremost搞不出来图片

		root@kali: ~/Downloads/misc higher/15	×	
<pre>root@kali:~/D</pre>	ownloads/misc	higher/15# binwalk 66b7f39d97364ca5	a2f928a4e61b46ee.pcap	
DECIMAL	HEXADECIMAL	DESCRIPTION		
0 4 5 5 6 26441 cuments 26577 () Downloads	0x0 0x6749 0x67D1	Libpcap capture file, little-end PNG image, 1754 x 2480, 8-bit/co Zlib compressed data, best compre	ian, version 2.4, Ethe lor RGBA, non-interlac ession	ernet, snaplen: 262144 ced
<pre>root@kali:~/D Processing: 6 [*] Pictures root@kali:~/D audit:txt root@kali:~/D O CentOS_6.5.</pre>	oownloads/misc 66b7f39d97364ca oownloads/misc	higher/15# foremost 66b7f39d97364ca 5a2f928a4e61b46ee.pcap higher/15# ls output/ higher/15#	5a2f928a4e61b46ee.pcap)
1 ++ // /> 200				

了解到有tcpxtract这个工具,Tcpxtract是一种基于文件签名从网络流量中提取文件的工具。安装tcpxtract,并使用分离图片出来,flag值出来了



名称	- 29% 🗸 🔹	0000000	00.png	Ľ,	= •	• •
© 0000000.png					属性	×
0000001.png				大小 : 类型	l754×248 PNG 图像	0 像素
66b7f39d97364ca5a2f928a4e61b46ee.pcap				文件大小 文件夹	L.0 MB L <u>5</u>	
output				光圈 曝光 焦距 ISO		
	HITB{95700d8a	efdc1648b90a92f3	3a8460a2c}	测光 相机		
				日期 时间		
	<		>			

16.hit-the-core

下载后是.core文件,.core文件是Linux的文件,用strings命令查看,看到一段特殊的文段



17.pure_color

下载后是一张图片,用stegsolve打开调试几下,即可看到flag

🛃 StegSolve 1.3 by Caesum		_		×
File Analyse Help				
Blue plane 0				
Flag	is			
true	vne heen treenb erednerenge	tor	nle	ļ
true_	_steganographers_doesnt_need_any_	to	ols	
true_	_steganographers_doesnt_need_any_	to (ols	•
true_	_steganographers_doesnt_need_any_	to	ols	•
true_	_steganographers_doesnt_need_any_	to	ols	

18.2-1

下载后是一张png图片,打开错误,用winhex打开,发现png文件头错误进行修正后发现还是打不开。

12 TOOT COTDUCTOR TOT DC0000000

Offset	0	1	2	3	4	5	6	7	8	9	А	В	С	D	Е	F	ANSI ASCII 🔺
00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	SPNG IHDR
00000010	00	00	00	00	00	00	02	F8	80	06	00	00	00	93	2F	8A	ø "/š
00000020	6B	00	00	00	04	67	41	4D	41	00	00	9C	40	20	0D	E4	k gAMA œ@ ä
00000030	CB	00	00	00	20	63	48	52	4D	00	00	87	0F	00	00	8C	Ë CHRM ‡ Œ
00000040	0F	00	00	\mathbf{FD}	52	00	00	81	40	00	00	7D	79	00	00	E9	ýR @}y é
00000050	8B	00	00	3C	E5	00	00	19	CC	73	3C	85	77	00	00	0A	< <å Ìs<…w
00000060	39	69	43	43	50	50	68	6F	74	6F	73	68	6F	70	20	49	9iCCPPhotoshop I
00000070	43	43	20	70	72	6F	66	69	6C	65	00	00	48	C7	9D	96	CC profile HÇ -
00000080	77	54	54	D7	16	87	CF	BD	77	7A	A1	CD	30	D2	19	7A	wTT× ‡Ï½wz;Í0Ò z
00000090	93	2E	30	80	F4	2E	20	1D	04	51	18	66	06	18	CA	00	".0€ô. Q f Ê
000000A0	C3	0C	4D	6C	88	A8	40	44	11	11	01	45	90	A 0	80	01	à Ml^"@D E €
000000B0	A3	Α1	48	AC	88	62	21	28	A 8	60	0F	48	10	50	62	30	£;H¬^b!("` H Pb0
000000000	8A	A 8	A 8	64	46	D6	4A	7C	79	79	EF	E5	E5	F7	C7	BD	Ś‴dFÖJ yyïåå÷ǽ
000000000	DF	DA	67	EF	73	F7	D9	7B	9F	в5	2E	00	24	4F	1F	2E	ßÚgïs÷Ù{Ÿµ. \$0 .
000000E0	2F	05	96	02	20	99	27	E0	07	7A	38	D3	57	85	47	D0	/ – ™'à z8ÓWGĐ
000000F0	В1	FD	00	06	78	80	01	A6	00	30	59	E9	Α9	BE	41	ΕE	±ý x€ ¦ OY驾Aî
00000100	C1	40	24	2F	37	17	7A	BA	C8	09	FC	8B	DE	0C	01	48	À@\$/7 z°Èü∢₽ H
00000110	FC	BE	65	E8	E9	4F	Α7	83	FF	4F	D2	AC	54	BE	00	00	ü¾eèéO§fÿOÓ¬T¾
00000120	C8	5F	C4	E6	6C	4E	ЗA	4B	C4	F9	22	4E	CA	14	A4	8A	E_ÄælN:KÄù"NÉ ¤S
00000130	ED	33	22	A6	C6	24	8A	19	46	89	99	2F	4A	50	C4	72	13"¦Æ\$S F‰™/JPAr
00000140	62	8E	5B	E4	A5	9F	7D	16	D9	51	CC	EC	64	1E	5B	C4	bZ[ä¥Y} UQIìd [A
00000150	E2	9C	53	D9	C9	6C	31	F7	88	78	7B	86	90	23	62	C4	âœSUEll÷^x{† #bĂ
00000160	47	C4	05	19	5C	4E	A6	88	6F	8B	58	33	49	98	CC	15	GA \N¦^o <x3i~i< td=""></x3i~i<>
00000170	F1	5B	71	6C	32	87	99	0E	00	8A	24	в6	0B	38	AC	78	ñ[ql2‡™ S\$¶ 8¬x
00000180	11	9B	88	98	C4	0F	0E	74	11	F1	72	00	70	A4	в8	2F	>°~A t ñr p¤,/
00000190	38	E6	0B	16	70	В2	04	E2	43	В9	A4	A4	66	F3	В9	71	8æ p² âC¹¤¤fó¹q

)这里涉及了png文件格式

IHDR. png															IDC	н				
Offset	0	文件	<mark>头</mark> 2	3	4	5	6	7	d	unik	LÀ	naih	C		B	Ma				
00000000	89	50	4E	47	OD	0A	1A	0A	00	00	<u>AU</u>	ΠD	49	48	44	52	I PNG		IHDE	2
00000010	ur la	Hit	02	9C	00	00	01	DD	08	06	ÔO	00	00	FE	1À	5A	1	Ý	þ Z	2
00000020	B6	00	00	00	04	73	42	49	54	08	08	08	08	7C	08	eh	t CCBB	IT	1 0	L
00000030	88	00	00	00	09	70	48	59	73	00	00	OB	12	00	00	OB	l pH	Ys		
00000040	12	01	D2	DD	7E	FC	00	00	00	16	74	45	58	74	43	72	ÒÝ~ü	1	tEXtCr	-
00000050	65	61	74	69	6F	6E	20	54	69	6D	65	00	31	32	2F	31	eation	Time	= 12/1	

(固定) 八个字节89 50 4E 47 0D 0A 1A 0A为png的文件头

(固定)四个字节00 00 00 0D (即为十进制的13)代表数据块的长度为13

(固定)四个字节49 48 44 52 (即为ASCII码的IHDR)是文件头数据块的标示 (IDCH)

(可变) 13位数据块 (IHDR)

前四个字节代表该图片的宽

后四个字节代表该图片的高

后五个字节依次为: Bit depth、ColorType、Compression method、Filter method、Interlace method

(可变)剩余四字节为该png的CRC检验码,由从IDCH到IHDR的十七位字节进行crc计算得到。

使用tweakpng这个工具计算crc校验码





使用脚本跑出正确的宽709,在windex修改成十六进制为02c5。

```
1
   import struct
 2 import binascii
 3 import os
 4
 5 m = open("misc4.png","rb").read()
 6 for i in range(1024):
 7
       c = m[12:16] + struct.pack('>i', i) + m[20:29]
 8
       crc = binascii.crc32(c) & 0xfffffff
 9
       if crc == 0x932f8a6b:
10
           print(i)
11
```



19.János-the-Ripper

下载压缩吧解压后不知道是什么文件,用file命令查看下是zip文件,改文件后缀名zip,解压需要密码,进行密码 爆破为fish,flag为flag{ev3n::y0u::bru7us?!}

20.2017_Dating_in_Singapore

题目为新加坡2017日历,附件解压打开时一张2017新加坡的日历,根据给出的数字进行连线,可得flag



Calendar for Year 2017 (Singapore)



21神奇的Modbus

下载下来是一个流量包,用wireshark打开,搜索关键字flag,追踪数据流,这里搜索关键字没用,全部查看下来 发现这里有flag, flag为**sctf{Easy_Modbus}**,要多加一个o

..... 6..... %P. \$..p... {.E.a.s.y. .M.d.b.u.s.}..}.c. 1. 2.R.%...

下载下来是一张图片,用binwalk查看一下,有点东西,foremost分离出来两张图片,用盲水印攻击,加密脚本: https://github.com/chishaxie/BlindWaterMark



23.can_has_stdio?

下载解压包解压得文件用记事本打开,看到得<,>,+,-,.,[,]等符号组成得五角星,可以猜测是Brainfuck语言了,

下面是这八种制	大态的描述,其中每个状态由一个字符标识:
字符	含义
>	指针加一
<	指针减一
+	指针指向的字节的值加一
-	指针指向的字节的值减一
	输出指针指向的单元内容(ASC II 码)
7	输入内容到指针指向的单元(ASC II 码)
[如果指针指向的单元值为零,向后跳转到对应的)指令的次一指令处
]	如果指针指向的单元值不为零,向前跳转到对应的[指令的次一指令处

(按照更节省时间的简单说法,"]"也可以说成"向后跳转到对应的"["状态"。这两解释是一样的。)

new 1 🗷 🔚 flag. txt 🛛 🔚 rockyou. txt 🗵 🔚 mi:	sc50🔀	
3	+	^
4	++	
5	+++	
б	++[>	
7	+>++>	
3	+++>++	
9	++>++++	
0	+>+++++	
1	>+++++>	
2	++++++>+	
3	++++++>++	
4 +	+++++>+++	
5 +	+++++>++++	
б ++++++>+++++++++++++++	+++++++++++>+++++++++++++++++++++++++++	
7 +++++++++++++++<<<<<<	<<<<<++<<	
3	>>>>>++++<<<<<<<	
9 >>>>>>+<<	<<<<<<>+<	
) (((((((((((((((((((((((((((((((((((((>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	
1 ~~~~~~~>>>>	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	
2 <<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	>>>>>>+++<	
3 ~~~~~~	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	
4 ~~~~~	<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	
5 >>	++++<<<<<<<<	
6 <>>>>>>	>>>>+<<<<	
7 <<<<>>>>>	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	
3 ~~~~~~	<<<>>>>>>>>	
9 +<<<<<	<<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	
) +++<<	<<<< <<<<>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	
1 >>>>>	+<	
2 >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	>>>++<	
3 ~~~~~	<<<>>>>>	
4 >>>>>	>>>+	
5 <<<<<	~~~~	
6 <<<	\Leftrightarrow	
7 >>>	>>	
3		
1 >>>>>++<<	>>>>>+<<<<<	

到https://www.splitbrain.org/services/ook这个网站翻译Brainfuck为文本

splitbrain.org/services	s/ook				G	☆ ७	2	Ø	(P)	\odot	69	hr	Ō	٥	R	٥	Q,
	Home	Projects	Services	Personal	Shop				Se	earc	ch						۹

BRAINFUCK/OOK! OBFUSCATION/ENCODING

This tool can run programs written in the <u>Brainfuck</u> and <u>Ook!</u> programming languages and display the output.

It can also take a plain text and obfuscate it as source code of a simple program of the above languages.

All the hard work (like actually understanding how those languages work) was done by Daniel Lorch and his Brainfuck interpreter in PHP

flag{esolang	gs_for_fun_and_profit}
Text to Ook!	Text to short OokI OokI to Text

24.5-1

考察xortool工具的使用,安装python库pip2 install xortool,猜测最有可能得密钥长度为13,

root@kali:~/Downloads/	misc higher/24# xortool=cd2a80e1b48e44b5a830605e684ffb31
The most probable key	lengths:containing the private key and the certificate
2: 12.2%	
-5:ach11d9% <dir></dir>	
-9:0-c9:8%-dir	
13: isa22e2%ip-versio	
20: 6.8%	
22: 6.2%	
26: 12.8%	
30:0-04.6%	
ro39:kal7:8%Downloads/	
Co52:cti5c7%ip	
Keyelengthican be 3*n/	
Most possible char is	needed to guess the key!4MB 677kB/s
root@kali:~/Downloads/	misc_higher/24# xortool cd2a80e1b48e44b5a830605e684ffb31 -l 13 -c 20
<pre>1 possibleikey(s)ioftl</pre>	engthol3:pip 18.1
Good tuck To Young pip-1	6.1:
Found 1 plaintexts wit	hs95.0%+ valid&characters
See files filename key	.csv,lfilename-char used-perc valid.csv
root@kali.~/Downloads/	misc higher/24# lsnut/zin/00000811/day2's secret# nin install opency-python

使用脚本解密出原文,flag为wdflag{You Are Very Smart}



he opening line of the novel famously announces: "It is a truth universally acknowledged, that a single man in possession of a good fortune must be in want of a wife." This sets marr tral subject—and really, a central problem—for the novel generally. Readers are poised to question whether or not these single men are, in fact, in want of a wife, or if such desire db y the "neighbourhood" families and their daughters who require a "good fortune". Marriage is a complex social activity that takes political economy, and economy more generally, i In the case of Charlotte Lucas, for example, the seeming success of her marriage lies in the comfortable economy of their household, while the relationship between Mr and Mrs Bennet llustrate bad marriages based on an initial attraction and surface over substance (economic and psychological). The Bennets' marriage is one such example that the youngest Bennet, Ly me to re-enact with Wickham, and the results are far from felicitous. wdflag(You Are Very Smart)Though the central characters, Elizabeth and Darcy, begin the novel as hostile acquain nlikely friends, they eventually work to understand each other and themsetves so that they can marry each other on compatible terms personally, even if their "equal" social status re t. When Elizabeth rejects Darcy's first proposal, the argument of only marrying when one is in loved. Elizabeth only accepts Darcy's proposal when she is certain she low or receiping are reciprocated. Austen's complex sketching of different marriage sultimately allows readers to question what forms of alliance are desirable, especially when it comes to accouncil a sumple attraction and barterable, especially when it comes to accouncil actual the autor of the relation of the relative set and the second and use in the accounce. Elizabeth only accepts Darcy's proposal when she is certain she low or accouncil actual the accounce. Austen's complex sketching of different marriages ultimately allows readers to question what forms of alliance are desirabl

25.MISCal

下载解压后不知道是什么文件,用file命令查看一下,是bzip2文件,修改文件后缀名为.bz2,解压文件

<pre>root@kali:~/Downloads/ e4a1278fef074ffd89b5bd root@kali:~/Downloads/</pre>	misc higher/25# 9b789527b5: bzip misc higher/25#	file e4a1278fef074ffd89b 2 compressed data, block tar -xjvf e4a1278fef074f	55bd9b789527b5 size = 900k fd89b5bd9b789527b	5.bz2
ctf/				
ctf/flag.txt ctf/.git/				
ctf/.git/description				
ctf/.git/refs/ ctf/.git/refs/heads/ ctf/.git/refs/heads/ma	Downloads/misc h ster			
<pre>ctf/.git/refs/stash ctf/.git/refs/tags/</pre>				
ctf/.git/ORIG_HEAD ctf/.git/logs/				
ctf/.git/logs/refs/				

发现是git目录,里面有个flag.txt文件不过flag值.git log查看git日志,git stash list列出所有保存的进度 列表,git stash apply恢复暂缓区的内容,有s.py文件,运行一下可以得到flag

<pre>root@kali:~/Downloads/misc higher/25# cd ctf/ root@kali:~/Downloads/misc higher/25/ctf# git log commit bea99b953bef6cc2f98ab59b10822bc42afe5abc (HEAD -> master) Author: Linus Torvalds <torvalds@klaava.helsinki.fi> Date: Thu Jul 24 21:16:59 2014 +0200</torvalds@klaava.helsinki.fi></pre>	<pre>linux1 x86 64.whl (28.7MB) Requirement already satisfied: numpy>=1.11.1 in /usr (from opencv-python) (1.14.5) Installing collected packages: opencv-python Successfully, installed opency-python-4.1.1.26</pre>
Initial commit root@kali:~/Downloads/misc higher/25/ctf# git stash list stash@{0}: WIP on master: bea99b9 Initial commit root@kali:~/Downloads/misc higher/25/ctf# git stash show flag.txt 25 ++++++ 2 files changed, 28 insertions(+), 1 deletion(-) root@kali:~/Downloads/misc higher/25/ctf# git stash apply 位于分支 master 要提交的变更:	
<pre>(使用 "git reset HEAD <文件>" 以取消留存) = 新文件: s.py 尚未暂存以备提交的变更: (使用 "git add <文件>" 更新要提交的内容) (使用 "git checkout <文件>" 丢弃工作区的改动)</pre>	
<pre></pre>	

26.3-1

下载下来是rar压缩包,不过文件没有文件后缀名,binwalk查看得rar,添加.rar文件后缀名,提取压缩包出来时 一个流量包,用wireshark打开,搜索flag关键字,追踪数据流,在第6个流可以看到flag.rar,base64编码

	Wireshark · 追踪 TCP	0	•	8
ζ件(E) 编辑(E) 视图(V) 跳转(G) 捕家	[root@localhost wireshark]# llss			
🚺 🗶 💿 to 📄 🖹 🚺	1 2 3 test			
tcp.stream eq 6	[root@localhost wireshark]# ccaatt 11			
分组列表 ▼ 宽窄	Rar!3			
o. Time Source	TU.<			
533 75.600871 10.1.70.				
534 /5.651030 10.1.10. 535 76.003148 10.1.10.	<u>19aaFYSQQKr+hVX6hl2smAUQ5a767TsULEUebWSajEo=</u> [root@localhost wireshark]# ppiinngg bbaaiidduuccoomm			
536 76.004238 10.1.70.	PING baidu.com (111.13.101.208) 56(84) bytes of data.			
537 76.054762 10.1.10. 538 76 148648 10 1 10	64 bytes from 111.13.101.208 (111.13.101.208): icmp_seq=1 ttl=48 time=33.4 ms			
539 76.149809 10.1.70.	64 bytes from 111.13.101.208 (111.13.101.208): icmp_seq=2 ttl=48 time=32.1 ms 64 bytes from 111 13.101.208 (111.13.101.208): icmp_seq=3 ttl=48 time=34.7 ms			
Frame 533: 126 bytes on wire	64 bytes from 111.13.101.208 (111.13.101.208): icmp_seq=4 ttl=48 time=31.9 ms			
Ethernet II, Src: Hangzhou_fb	^C			
Transmission Control Protocol	4 packets transmitted, 4 received, 0% packet loss, time 3037ms			
Telnet	rtt min/avg/max/mdev = 31.921/33.067/34.784/1.155 ms			
	[root@localnost wiresnark]# ccaalt 33			
	# coding:utf-8			
000 ec a8 6b 3a c1 e5 80 f6				
010 00 70 e5 f6 40 00 3f 06				
030 00 e5 67 da 00 00 31 39	author = 'YFP'			
040 72 2b 68 56 58 36 68 6c				
060 6f 3d 5b 72 6f 6f 74 40				
070 74 20 77 69 72 65 73 68	from Crypto import Random			
	from Crypto.Cipher import AES			
	*			
	import sys			Ŧ
	107 客户簿分组, 121 服务器 分组, 202 turn(s).	_		
	Entire conversation (2,696 bytes)	ì	充 6	-
	查找:	查找下	一个(<u>N</u>)
	2Help 速相供流 ITFI Save as 该		Clos	

导出对象,选择http,flag.rar文件,解压需要密码,可以从数据流中可以看解密脚本多半时压缩包密码

	Wireshark · 追踪 TCP 流 (tcp.stream eq 6) · ++++					
文件(E) 编辑(E) 视图(⊻) 跳转(G) 捕获					Т	
a 🗉 🧟 💿 🖨 🖺 🧕	# coding:utf-8					
tcp.stream eq 6					I	
分组字节流 ▼ 宽窄	author = 'YFP'				ſ	
o. Time Source					I	
639 85.083171 10.1.10.					T	
640 85.084032 10.1.70.					1	
644 85.134077 10.1.10.	from Crypto import Random					
645 85.411247 10.1.10.					1	
646 85.412/83 10.1.70.	from Crypto.Cipher Import AES					
648 85,463927 10.1.70						
Frame 648: 601 bytes on wire					E	
Ethernet II Src: Hangzhou fh	import sys					
Internet Protocol Version 4.						
Transmission Control Protocol	import base64					
Telnet						
Data: # coding:utf-8\r						
Data: \r\n						
Data: \r	IV = 'QWERTYUIOPASDFGH'				1	
0030 00 e5 01 08 00 00 23 20					F	
0040 74 66 2d 38 0d 00 0d 0a					1	
0050 74 68 6f 72 5f 5f 20 3d					1	
0060 Od Oa Od O0 Od Oa 66 72	def decrypt(encrypted):				1	
0070 6f 20 69 6d 70 6f 72 74	•				1	
000 00 00 00 00 72 01 00 20	aes = AES.new(IV, AES.MODE_CBC, IV)				1	
00a0 Od 00 Od 0a 0d 00 Od 0a					1	
00b0 79 73 0d 00 0d 0a 69 6d	return aes.decrypt(encrypted)				1	
00c0 65 36 34 0d 00 0d 0a 0d					1	
00d0 27 51 57 45 52 54 59 55					1	
0000 48 27 00 00 00 0a 00 00 00f0 63 72 79 70 74 28 65 60	(df.anchut/mersage)					
100 3a 0d 00 0d 0a 20 20 61	uer energy (inessage).					
0110 2e 6e 65 77 28 49 56 2c	U			Ŧ		
0120 45 5f 43 42 43 2c 20 49	107 客户端分组, 121 服务器分组, 202 turn(s).				1	
0130 72 65 74 75 72 6e 20 61				T	1	

提取出来整合修改下

```
# coding:utf-8
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 = '19aaFYsQQKr+hVX6hl2smAUQ5a767TsULEUebWSajEo='
example = decrypt(base64.b64decode(str))
print example
print decrypt(example)
```

root@kali:~/Downloads// passwd={No_One_Can_Dec v~VeVaVVVVfVVV,q 0?09>	misc higher/26# python test.py rypt Me} 解且感問 SZ@c@	oot@kali
打开(0) 👻 🖪	flag.txt ~/.cache/.fr-VlyqL0	保存(S) =
WDCTF{Seclab_CTF_2017}		

27.适合作为桌面

下载压缩包解压是一张图片,放stegsolve弄一下,可以看到一个二维码,扫一下可以看到一串数字

File Analyse Help	
Red plane 1	
	QR Research $-\Box \times$
当 本 1 3 3 3 3 3 3 3 3 3 3 3 3 3	错等级:L,掩码:3]容: F30D0479C8055863000000000000000000000000000000000
05	32802000006300000000300000016000000430000007378000000640100640200640300640400640
50	00640600640700640300640800640900640A00640600640B00640A00640700640800640C00640C0064 000640E00640900640F006716007D00006410007D0100781E007C0000445D16007D02007C0100740
00	07C0200830100377D0100715500577C010047486400005328110000004E6966000000696C00000069
61	0000006957000000697000006933000000693800000693500000693700000069300000693200 0069340000006931000006965000000697D0000007400000002801000007403000006368722803
00	000074030000007374727404000000666C6167740100000069280000000280000000730400000031
2E 00 00	/0/952030000000028000000730A000000001480106010D0114014E280100000052030000002800000 028000000002800000007304000000312E707974080000003C6D6F64756C653E0100000073000000
「日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日本の日	<u> 冯完成</u>

这是一串十六进制的数字,在winhex里创建一份文件,把这些数字写进去,保存为.pyc

🚟 WinHex - [test.pyc]

🧱 文件(E) 编辑(E) 搜索(S) 导航(N) 查看(V) 工具(I) 专业工具(I) 选项(Q) 窗口(W) 帮助(H)

案件数据	🗅 🍯 🖬 😃 🍋	🤹 😭 🛸	- *) 🖣	a 🖽 G	B 1012	2	мм,	HEX 😘	HEX	-	-101 •	← →		🍣 🌫 🔅 🥅 🔎 🧠 🏠 🏭
文件(L) 编辑(D)	test.pyc													
	Offset	0 1	2 3	4	56	7	8	9 A	В	С	D	Е	F	ANSI ASCII
	00000000	03 F3 0	D OA	79 c	CB 05	58	63 0	0 00	00	00	00	00	00	ó yË Xc
	00000010	00 01 0	00 00	00 4	0 00	00	00 7	'3 OD	00	00	00	64	00	@sd
	00000020	00 84 0	00 00	5A 0	00 00	64	01 0	0 53	28	02	00	00	00	"ZdS(
	00000030	63 00 0	00 00	00 0	00 80	00	00 1	6 00	00	00	43	00	00	c C
	00000040	00 73 7	78 00	00 0	0 64	01	00 6	54 02	00	64	03	00	64	sx d d d d
	00000050	04 00 6	54 05	00 6	54 06	00	64 0	07 00	64	03	00	64	80	d d d d
	00000060	00 64 0	09 00	64 0	00 A(64	06 0	0 64	0B	00	64	0A	00	d d d d
	00000070	64 07 0	00 64	08 0	0 64	0C	00 6	54 OC	00	64	0D	00	64	ddddd
	00000080	0E 00 6	54 09	00 6	54 OF	00	67 1	6 00	7D	00	00	64	10	ddg}d
	00000090	00 7D 0	01 00	78 1	E 00	7C	00 0	0 44	5D	16	00	7D	02	} x D] }
	000000A0	00 7C 0	01 00	74 0	00 00	7C	02 0	0 83	01	00	37	7D	01	t f 7}
	000000в0	00 71 5	55 00	57 7	/C 01	00	47 4	8 64	00	00	53	28	11	qUW GHd S(
	000000000	00 00 0	00 4E	69 6	56 00	00	00 6	59 6C	00	00	00	69	61	Nif il ia
	00000000	00 00 0	0 69	67 0	00 00	00	69 7	'B 00	00	00	69	33	00	ig i{ i3
	000000E0	00 00 6	59 38	00 0	00 00	69	35 0	00 00	00	69	37	00	00	i8 i5 i7
	000000F0	00 69 3	30 00	00 0	0 69	32	00 0	00 00	69	34	00	00	00	i0 i2 i4
	00000100	69 31 0	00 00	00 6	59 65	00	00 0	0 69	7D	00	00	00	74	il ie i} t
	00000110	00 00 0	00 00	28 0	01 00	00	00 7	4 03	00	00	00	63	68	(t ch
	00000120	72 28 0	03 00	00 0	00 74	03	00 0	00 00	73	74	72	74	04	r(t strt
	00000130	00 00 0	00 66	6C 6	51 67	74	01 0	00 00	00	69	28	00	00	flagt i(
	00000140	00 00 2	28 00	00 0	00 00	73	04 0	00 00	00	31	2E	70	79	(s 1.py
	00000150	52 03 0	00 00	00 0	01 00	00	00 7	'3 OA	00	00	00	00	01	R s
	00000160	48 01 0	06 01	0D 0	01 14	01	4E 2	8 01	00	00	00	52	03	H N(R
	00000170	00 00 0	00 28	00 0	00 00	00	28 0	00 00	00	00	28	00	00	(((
	00000180	00 00 7	73 04	00 0	00 00	31	2E 7	0 79	74	08	00	00	00	s 1.pyt
	00000190	3C 6D 6	5F 64	75 6	5C 65	3E	01 0	00 00	00	73	00	00	00	<module> s</module>
	000001A0	00												

到https://tool.lu/pyc/这个网站解密pyc文件,解密出来python脚本跑一下即为flag



28.Banmabanma

下载下来是一张斑马的图片,猜测是条形码



到https://online-barcode-reader.inliteresearch.com/解码一下,都不用ps一下了, nb

Free Online Ba To get such results u If your business ap email your technic email your sales in	rcode Reader sing ClearImage SDK use TBR Code 103. plication needs barcode recognition capabi al questions to support@inliteresearch.com quiries to sales@inliteresearch.com	lities,	Barcode Reader Software Development Kit (SDK). Decode barcodes in C#, VB, Java, C\C++, Delphi, PHP and other languages. Get ClearImage SDK
File: 斑马斑马 Pages: 1 Barcode: 1 of 1 Length: 16 Module: 1.6pix FLAG IS TENSHINE	.png Type: Code39 Rotation: none Rectangle: {X=71,Y=93,Width=410,He	New File Barcodes: 1 Page 1 of 1 eight=119}	Barcode Director. Barcode scanner application renames, sorts and splits documents using barcode values. Get Barcode Director Barcode Reader Web Server with RESTful API. Client SDKs for JavaScript, .NET (C# or VB), Java, Node.js, PHP, Python or Ruby. Web API Test Server This site offers free limited demonstration. See terms of service.

29.simple_transfer

下载下来是个流量包,binwalk有点东西,foremost分离出来个pdf,打开pdf即可看到flag

	▲ 主文件夹 / Downloads / misc higher 29 / output / pdf -
名称	
00000662.pdf	
□ 1 共1页	▶ 00000662.pdf 56.7% ■ ■ ●

30.warmup

下载下来是一张图片和一个加密的zip包,加密的zip也有一个open_forum.png的文件,把open_forum.png压缩用 ARCHPR明文进行攻击,破解出加密的zip包,这里要注意压缩软件要使用winrar

		···-		
🕵 open_forum.png	2017/12/6 12:06	图像 (png) 文件 4	2 KB	
욝 open_forum.zip	2019/10/7 23:20	加密密钥已成功恢复!		\times
Same and the second	2017/12/6 12:06	Advanced Arch	nive Password Recovery 统计信息:	
-		总计口令	n/a	_
🔏 ARCHPR 4.54 - 100%	—	总计时间	12s 928ms	
文件(F) 恢复(R) 帮助(H)		平均速度(口令/秒)	n/a	
	1 0	这个文件的口令	未找到	2
打开 开始 偏止 基准测试	↓ 43 ↓ 43 ↓ 43 ↓ 43 ↓ 43 ↓ 43 ↓ 43 ↓ 43 ↓ 43 ↓ 43 ↓ 43 ↓ 43 ↓ 43 ↓ 43 ↓ 44 ↓ 44 ↓ 44 ↓ 44 ↓ 44 ↓ 44 ↓ 44 ↓ 44 ↓ 44 ↓ 44 ↓ 44 ↓ 44 ↓ 44 ↓ 44 ↓ 44 ↓ 44	加密密钥	[e43a642a 8e424cdb eb7c6331]	9
加密的 ZIP/RAR/ACE/ARJ 文件	攻击类型		✔ 确定	
C:\Users\jaie\Desktop\1\warmup_3D8711	明文	~		
 范围 长度 字典 明文 自动保存 明文选顶 明文文件路径: C:\Users\jaie\Desktop\1\open_forum.zip 密钥 e43a642a 密钥 8e 允许使用二进制文件作为明文 ZIP 档案文 状态窗口 2019/10/7 23:26:53 - 文件*C:\Users\jaie\Desktip\1\varymp_30871198 IPD69603E778A1292A002019/10/7 23:26:53 - 明文或击已开始 2019/10/7 23:26:53 - 明文或击已开始 2019/10/7 23:27:06 - 加密密钥已成功恢复! 当前口令: n/a 可 明文或击正在进行,尝试找回可打印口令(最 100% 	选项 高级 → 开始于: 0 +424cdb 密钥 eb7c63 Z件 top 07C48.zip"已打开。 平均速度: n/a 刺余时间: 0s 长 10 个字符) 6	31		темр

解压出来的是两张一样的图,猜测是盲水印,这里使用https://github.com/chishaxie/BlindWaterMark解盲水印的脚本跑出水印的图片即为flag



31.我们的秘密是绿色的

下载下来是张日历图片,这里根据题目名字提示,我们要用到Our secret文件隐藏加密软件,密码即是日历图中 绿色数字0405111218192526,



Our Sec	ret		Make you	r secrets inv	visible in just 3 ec	isy steps !				
\square	HIDE			UNHIDE						
T	Step 1:	Select a carrier file		Step1: S	pecify a carrier file	•				
	我们的	N秘密是green.jpg S	ize: 260610	我们的秘	密是green.jpg Si	ze: 260610 bytes				
	Step 2:	Add/remove file or n	nessage	Step 2: E	nter password					
	T A	dd BRemov	e	•••••	•••••					
	Туре	Name	Size (k)	🆄 Ur	hide (doul	ble click to save)				
Help				Туре	Name	Size (k)				
				File	try.zip	1				
About										
	<		>							
Exit	Step 3: I	Password								
	Enter pa	ssword again		<		>				
		(📝 Hide							

可以得到try压缩包,用记事本打开可以看到有生日提示,那就用生日当作密码进行暴力破解,密码最长8位

pd%葃X?塄?□

December and a contract and a contr

	🔏 ARCHPR 4.54 - 1	9%	_		\times
	文件(F)恢复(R) 幕	署助(<u>H</u>)			
	00 -0				
口令已成切恢复	륀.		×	¥T.	19.00
	Advanced Archive Pa	ssword Recovery 统计信息:		大丁	лец.
总计口令		19,950,821			
总计时间		829ms			\sim
平均速度(口令	☆/秒)	24,066,129			
这个文件的口	l令	19950822	6		
十六进制口令	÷	31 39 39 35 30 38 32 32	6		N
	- 保存	✔ 确定			•
	 □ 空格 □ 所有可打印字符 □ 状态窗口 	〕HB,H	用	1户定义 🗌	aB Co
	2019/10/8 11:28:57 - 2019/10/8 11:28:57 - 2019/10/8 11:28:58 - 2019/10/8 11:28:58 -	· 开始于口令 '00000000' · 开始暴力攻击 · 口令已成功恢复! ' 19950822' 是这个文件的一个有效口令			<
	当前口令: 199 已用时间: 口令长度 = 8, 总计: 1	50822 平均速度: 剩余时间: 01,111,109,已处理: 19,950,821	24, 1 3s	82,813 p/s	
		19%			_
	ARCHPR version 4.54 (c)) 1997-2012 ElcomSoft Co. Ltd.			

有破解的密码解压还是一个带加密的zip包,不过zip也有一个readme.txt,明文攻击走起,得到密码Y29mZmVI

	음 flag.	zip			2017/4/20 15:24		ZIP 压缩文件		1 KB				
	📔 read	me.txt			2017/4/20 14:59		TXT 文件		1 KB				
🔇 flag.	zip - Bandizip	o 6.10											
文件(E)	编辑(<u>E</u>) 查抄	^{讫(])} 选项(<u>O</u>)	视图(⊻) 帮	助(<u>H</u>)				æ	ARCHPR 4.54 -	0%			\times
		- G						A	口令已成功恢复!				×
\rightarrow		f ~	Į	+		- 57	:	Æ		Advanced Archive F	assword Recover	ry 统计信息:	_
打开	解	压	新建	添加	删除	测试	查看	代码	总计口令		n/a		
🔒 flag.z	tip		名称		0				总计时间		2m 15s 306ms		_
	-		📔 readme	.txt*					平均速度(口令/利	9)	n/a		- 0-
			🔝 flag.zip*	ŧ					这个文件的口令		Y29mZmVI	61 F6 6	
								l l	十六进制口令		59 32 39 6d 5a	a 60 56 6C	
									- 1	保存		✔ 确定	
										1630eea 密钥	49573262	密钥 1092e65c	1
									1 允许使用二进制	」文件作为明文 ZIF	档案文件		
									状态窗口				
								ſ	2019/10/8 11:39:48	- 明文攻击已开始			^
									2019/10/8 11:42:03) - 加密密钥已成功) - 口令已成功恢复	伙夏! 		
									2019/10/8 11:42:03	3 - 'Y29mZmVl' 是这·	个文件的一个有刻	效口令	~
									í前口令: n/ 1用时间: 2m 1文攻击正在进行,	'a n 15s 尝试找回口令(最	平均速度: 剩余时间: 长9个符号)	n/a 15s	
											0%		

解压后又一个加密压缩包,爆破无解,用winhex打开看看,看到伪加密了,把01改成00,解压就不需要密码了

test.pyc	flag.z	ip																		
Offset	t	0	1	2	3	4	5	6	7	8	9	А	В	С	D	Е	F		ANSI AS	SCII
000000	00	50	4B	03	04	14	00	00	80	08	00	66	76	94	4A	7D	AF	PK	fv'	″J}¯
000000	10	72	9F	1E	00	00	00	1E	00	00	00	80	00	00	00	66	6C	rŸ		f]
0000002	20	61	67	2E	74	78	74	2в	4C	49	29	28	2C	\mathbf{CF}	2в	48	2E	ag.tx	t+LI)(,İ	Ċ+Η.
000000	30	C8	49	$^{\rm CD}$	53	2D	28	02	в2	E3	AB	E3	AB	AA	в4	52	1C	ÈIÍS-	(²ã«ã«'	'R
0000004	40	D2	0в	6В	01	50	4B	01	02	3F	00	14	00	01	09	80	00	ÒkPI	K ?	
000000	50	66	76	94	4A	7D	AF	72	9F	1E	00	00	00	1E	00	00	00	fv"J}	rΫ	
000000	60	80	00	24	00	00	00	00	00	00	00	20	00	00	00	00	00	\$		
000000	70	00	00	66	6C	61	67	2E	74	78	74	0A	00	20	00	00	00	fla	g.txt	
000000	80	00	00	01	00	18	00	E6	\mathbf{FC}	D6	7E	A2	в9	D2	01	2C	E6		æüÖ∼¢¹(ò,a
000000	90	57	65	82	в9	D2	01	2C	E6	57	65	82	в9	D2	01	50	4B	We,1Ò	,æWe,¹(Ò PF
0000002	A0	05	06	00	00	00	00	01	00	01	00	5A	00	00	00	44	00		Z	D
0000001	в0	00	00	00	00															

解压后得到一个flag.txt文本,内容为qddpqwnpcplen%prqwn{zz*d@gq},猜测栅栏密码,再凯撒密码

Crypto	Image UnZip
填写所需	检测的密码:(已输入字符数统计:30)
qddpqwn	pcplen%prqwn_{_zz*d@gq}
结果:(台	字符数统计: 234)
得到因数	(排除1和字符串长度):
2350	6 10 15
第1栏: c	qdqnclnpqn{z*@qdpwppe%rwzdg}
第2栏: c	<pre>qpnpnrn_*gdqpl%q_zdqdwcepw{z@}</pre>
第3栏: c 第1栏: c	gwlr{ddneq_@dpnwzgpc%nzqqpp_*}
第4栏: C 第5栏: c	al{de_dnzp%zap*wrdna@pwgcnap_}
第6栏: c	grdqdwpnq_w{n_pzczp*lde@ng%qp}

Crypto	Image	UnZip			
填写所需检	金测的密码	:(已输	入字符数统计: 30)	
qwlr{ddne	eq_@dpnwa	zgpc%nzq	abb_*}		
纣里·(字	符粉结计	• 780)			
······································	11 30 2007	. 7007			
rxms{eeot	r_@eqoxa	ahqd%oar	rqq_*}		
synt (TTpg + zou / ccck	gs_@trpy: 	oire%pbs sisf%ast	srr_*j		
uany (bhri	iu_@gsqzo	lk+a%rdu	uss_*; .++ *l		
vbow{iis	iv @iusbe	aluh%sev	/uu *}		
wcrx{iit	w @ivtc	fmvi%tfw	vuu_) vvv *}		
xdsy {kkul	lx_@kwudg	gnwj%ugx	<ww_*}< th=""><td></td><td></td></ww_*}<>		
yetz{ vn	ny_@lxvel	noxk%vhy	/xx_*}		
zfua{mmwr	nz_@mywf	ipyl%wiz	zyy_*}		
agvb{nnxc	ba_@nzxg	jqzm%xja	azz_*}		
bhwc{ooyp	ob_@oayhl	kran%ykb	baa_*}		
cixd{ppzc	qc_@pbzi	lsbo%zlc	cbb_*}		
djye{qqar	rd_@qcajr	ntcp%amd	dcc_*}		
ekzt{rrbs	se_@rdbkr	nudq%bne	edd_*}		
riag (ssci	ur_@secio	over%cor	ree_*;		
hnci (uuev	ig_etiamp /h_@ugen/	ixgt%eab	יי_*) אסס *}		
iod i {vvfv	vi @vhfo	vhu%fri	ihh *}		
ipek (wwg)	(j @wigps	sziv%gsi	;;; *}		
kqfl{xxhy	/k_@xjhqt	tajw%htk	<jj_*}< th=""><td></td><td></td></jj_*}<>		
lrgm{yyiz	zl_@ykiru	ubkx%iul	lkk_*}		
mehn (zzie	am @zliev	velv% ivm	n *}		

32.Just-No-One

下载下来后是一个.exe程序,安装后提醒flag值再安装许可协议,所以找吧,flag即为ILOVEREADINGEULAS

: 🛃	Setup - Binathlon 10 —	×
	License Agreement Please read the following important information before continuing.	
	Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.	5
	OF THE POSSIBILITY OF SUCH DAMAGES, OR ANY CLAIM BY A THIRD PARTY. RENTAL. YOU MAY NOT LOAN, RENT, OR LEASE THE SOFTWARE. 7A. YOU MAY SUBMIT THIS TO GET TEN POINTS: ILOVEREADINGEULAS . 8. UPGRADES. IF THE SOFTWARE IS AN UPGRADE FROM AN EARLIER RELEASE OR PREVIOUSLY RELEASED VERSION, YOU NOW MAY USE THAT UPGRADED PRODUCT ONLY IN ACCORDANCE WITH THIS EULA. IF THE BINATHLON 10 SOFTWARE PRODUCT IS AN UPGRADE OF A SOFTWARE PROGRAM WHICH YOU LICENSED AS A SINGLE PRODUCT, THE BINATHLON 10 SOFTWARE PRODUCT MAY BE USED ONLY AS PART OF THAT SINGLE PRODUCT PACKAGE AND MAY NOT BE SEPARATED FOR USE ON MORE THAN ONE COMPUTER. 9. OEM PRODUCT SUPPORT. PRODUCT SUPPORT FOR THE BINATHLON 10 SOFTWARE PRODUCT	7.
	◯ I <u>a</u> ccept the agreement	
-	I do not accept the agreement	
	< <u>B</u> ack <u>N</u> ext >	Cancel

33.Erik-Baleog-and-Olaf

下载下来是一张图片,用winhex打开看看,可以看到一个图片网址,http://i.imgur.com/22kUrzm.png

到这个网址把图片下载下来, (需要fq)再对比两张图片, 这里我用到的软件是beyond compare,可以看到二维 码

		_															- · · · · · ·
test.pyc flag.zip 0000000.png																	
Offset	0	1	2	3	4	- 5	6	7	8	9	Α	В	С	D	Е	F	ANSI ASCII 🔺
00006760	21	84	10	42	08	21	84	10	42	08	21	84	10	42	08	21	!" B !" B !" B !
00006770	84	10	42	08	21	84	10	42	08	21	84	10	42	08	21	84	" B ! " B ! " B ! "
00006780	10	42	08	21	84	10	42	80	21	84	10	42	80	21	84	10	в!"в!"в!"
00006790	42	08	21	84	10	42	08	21	84	10	42	08	21	84	10	42	В!"В!"В!"В
000067A0	08	21	84	10	42	08	21	84	10	42	08	21	84	10	42	80	!"B!"B!"B
000067B0	21	84	10	42	08	21	84	10	42	80	21	84	10	42	80	21	!" B !" B !" B !
000067C0	84	10	42	08	21	84	10	42	08	21	84	10	42	08	21	84	"В!"В!"В!"
000067D0	10	42	80	21	84	10	42	80	21	84	10	42	80	21	84	10	в!"в!"в!"
000067E0	42	80	21	84	10	42	08	21	84	10	42	80	21	84	10	42	В!"В!"В!"В
000067F0	08	21	84	10	42	08	21	84	10	42	80	21	84	10	42	08	!" B !" B !" B
00006800	21	84	10	42	08	21	84	10	42	08	21	84	10	42	08	21	!" B !" B !" B !
00006810	84	10	42	08	21	84	10	42	08	21	84	10	42	08	21	84	"В!"В!"В!"
00006820	10	42	08	21	84	10	42	08	21	84	10	42	08	21	84	10	в!"в!"в!"
00006830	42	80	21	84	10	42	08	21	84	10	42	80	21	84	10	42	В!"В!"В!"В
00006840	08	21	84	10	42	08	21	84	10	42	80	21	84	10	42	08	!" B !" B !" B
00006850	21	84	10	42	08	21	84	10	42	08	21	84	10	42	08	21	!" B !" B !" B !
00006860	84	10	42	08	21	84	10	42	08	21	84	10	42	08	21	84	"В!"В!"В!"
00006870	10	42	08	21	84	10	42	08	21	84	10	42	08	21	84	10	в!"в!"в!"
00006880	42	08	21	84	10	42	08	21	84	10	42	80	21	84	10	42	В!"В!"В!"В
00006890	08	21	84	10	42	08	21	84	10	42	08	21	84	10	42	08	!"B!"B!"B
000068A0	21	C4	BB	E7	$\mathbf{F}\mathbf{F}$	07	13	EC	56	32	A2	$\mathbf{F}\mathbf{F}$	D8	6C	00	00	!Ä»çÿ ìV2¢ÿØl
000068B0	00	23	74	45	58	74	68	69	6E	74	00	68	74	74	70	ЗA	#tEXthint http:
000068C0	2F	2F	69	2E	69	6D	67	75	72	2E	63	6F	6D	2F	32	32	//i.imgur.com/22
000068D0	6В	55	72	7A	6D	2E	70	6E	67	0E	AF	FD	3E	00	00	00	kUrzm.png ⁻ ý>
000068E0	00	49	45	4E	44	AE	42	60	82								IEND@B`,



34.Py-Py-Py

35.Reverse-it

下载下来一个不知道是什么文件,binwalk也看不出来,放进winhex看一下,可以文件末尾数字反转过过来就是 jpg的文件头FFD8FF。再结合题目名字,把所有的数字反转过来,winhex新建一个文件,保存jpg

格式

Offset	0	1	2	3	4	5	6	7	8	9	А	в	С	D	Е	F		AN	SI A	SCII
0001C60	D2	66	46	27	D2	23	23	F2	23	03	F2	93	93	93	13	F2	ÒfF'	Ò##ò	# ò"	ò ""
0001C70	76	27	F6	E2	33	77	E2	77	77	77	F2	F2	A3	07	47	47	v'öâ	3wâw	wwòò	£ GG
0001C80	86	22	D3	66	46	27	A3	37	E6	C6	D6	87	02	64	44	25	†"óf	F'£7	æÆÖ‡	: dD%
0001C90	A3	66	46	27	C3	02	E3	22	03	E2	43	E2	53	02	56	27	£fF'	Ãã"	âCâ	s v'
0001CA0	F6	34	02	05	D4	85	22	D3	в6	47	07	D6	87	A3	87	02	ö4 -	ô"ó	¶GÖ)‡£‡
0001CB0	22	F2	16	47	56	D6	A3	37	E6	A3	56	26	F6	46	16	22	"ò G	VÖ£7	æ£V&	öF "
0001CC0	D3	87	A3	37	E6	C6	D6	87	02	16	47	56	D6	07	D6	87	Ó‡£7	±ÆÖ‡	GV	7Ö Ö‡
0001CD0	A3	87	C3	02	E3	F3	22	46	93	36	в6	Α7	36	45	E4	A7	£‡Ã	ãó"F	``6¶§	6Eä§
0001CE0	35	56	27	Α7	84	96	86	56	34	07	D4	03	D4	53	75	22	5V'§	"-†V	4 Ô	ÔSu"
0001CF0	D3	46	96	02	22	\mathbf{FB}	BB	FE	22	D3	E6	96	76	56	26	02	ÓF–	"û»þ	"Óæ-	•vV&
0001D00	47	56	в6	36	16	07	87	F3	C3	00	F2	03	E2	13	F2	07	GV¶6	‡ó.	Ãò	âò
0001D10	16	87	F2	D6	F6	36	E2	56	26	F6	46	16	E2	37	E6	F2	‡òÖ	ö6âV	&öF	â7æò
0001D20	F2	A3	07	47	47	86	09	90	1E	$\mathbf{F}\mathbf{F}$	00	00	00	00	A1	00	ò£ G	Gt	Ÿ	i i
0001D30	00	00	10	00	00	00	40	00	30	0A	8C	00	00	00	10	00		0	0 Œ	
0001D40	00	00	40	00	20	0A	00	00	10	00	10	00	00	00	30	00	0			0
0001D50	10	0A	30	00	00	23	03	A3	13	43	A3	03	13	02	23	23	0	# £	C£	##
0001D60	A3	03	03	A3	13	03	03	23	64	00	14	46	F6	26	56	02	££	#	d F	′ö&V
0001D70	05	86	F6	47	F6	37	86	F6	07	02	54	C6	56	D6	56	E6	†öG	ö7†ö	TA	₹VÖVæ
0001D80	47	37	10	00	00	00	84	00	00	00	10	00	00	00	84	00	G7	"		
0001D90	00	00	00	00	00	00	0A	00	00	00	10	00	00	00	40	00				0
0001DA0	96	78	C8	00	00	00	41	00	00	00	20	00	23	10	27	00	-xÈ	А		# '
0001DB0	00	00	91	00	00	00	20	00	13	10	00	00	20	00	10	00	× 1			
0001DC0	00	00	30	00	82	10	A 6	00	00	00	10	00	00	00	50	00	0	,		P
0001DD0	в1	10	26	00	00	00	10	00	00	00	50	00	A1	10	00	00	ے ±		P	i
0001DE0	10	00	10	00	00	00	30	00	21	10	70	00	80	00	00	00		0	! p	€
0001DF0	A2	00	D4	D4	00	00	66	96	87	54	2D	00	1E	$\mathbf{F}\mathbf{F}$	00	00	¢ÔÔ	f-	‡т-	Ÿ
0001E00	84	00	84	00	10	10	10	00	64	94	64	A4	01	00	0E	FF			d″d¤	Ϋ́
0001E10	8D	FF														_	Ÿ			

SECCON{6in_tex7}

36.mysql

下载出来一个压缩包,再解压后的所有文件内容搜索关键字flag,再ib_logfile0文件找到flag,



37.4433

38.4433

40.心仪的公司

下载解压时一个流量包,追踪http流可以看到flag,有点难找啊

Inc. Take Source Destination Previous 1. 161.964521 192.168.1.108 192.168.1.101 CC 1		分组字节流 > 宽窄	~ □ 区分大小写	字符串 > f14	6cT.~4 S.e)=.U1}.i.d.
$ \begin{bmatrix} 1. 161.964498 \\ 192.168.1.111 \\ 192.168.1$	No.	Time	Source	Destination	r. txjoaL EM%7.][!a).Q.,
$ \begin{bmatrix} 1. 161.964521 \\ 1. 161.964521 \\ 1. 161.964644 \\ 192.168.1.111 \\ 192.168.1.108 \\ 192.168.1.108 \\ 192.168.1$	1	. 161,964498	192.168.1.108	192.168.1.111	c 2.h
1 1.161.964644 192.168.1.111 192.168.1.108 HT 1.161.969293 192.168.1.111 192.168.1.108 (-*, -, *, 2, 2, ·, ., 1, 2, ·, ., -, 2, ·, ., ., (-*, ., -, 4, ., ., ., -, ., ., -, ., ., ., ., ., ., ., ., ., ., ., ., .,	1.	. 161,964521	192.168.1.111	192,168,1,108	off
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	1.	. 161,964644	192.168.1.111	192,168,1,108	۲٫۰۰٬۰٬۰٬۱٬۰٬۲٬۲٬۲٬۲٬۲٬۲٬۲٬۲٬۲٬۲٬۲٬۲٬۲٬۲٬۲
1.161.969715 192.168.1.111 192.168.1.108 T w*.*.B.JUTNO.1 cxx2(.kq.t.s., FakqB.yx), SC(S]P.s.;/ 1.161.977939 192.168.1.111 192.168.1.108 T 1.161.977939 192.168.1.108 192.168.1.108 T 1.161.977939 192.168.1.108 192.168.1.108 T 1.161.977939 192.168.1.108 T T 1.161.977939 192.168.1.108 T T 1.161.978147 192.168.1.108 T T 1.161.978147 192.168.1.108 T T 1.161.978145 192.168.1.108 T T 1.161.978145 192.168.1.108 T T 1.161.978145 192.168.1.101 T T <	1.	. 161,969693	192.168.1.111	192.168.1.108	the second se
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	1.	. 161,969715	192.168.1.111	192,168,1,108	<pre>C W**B.jUTh0.1 <xmz(.&q.f6.farq8.yx?scqs]p.s.;.< pre=""></xmz(.&q.f6.farq8.yx?scqs]p.s.;.<></pre>
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	1	161.969721	192.168.1.111	192.168.1.108	
1 1.161.977995 192.168.1.111 192.168.1.118 192.168.1.118 192.168.1.111 192.168.1.118 192.168.1.111 <t< td=""><td>1.</td><td> 161,977939</td><td>192.168.1.108</td><td>192.168.1.111</td><td>C i 'B [[6*2.2S "7 E N 6#/ n · [* V II] e[* / OV II] hu?</td></t<>	1.	161,977939	192.168.1.108	192.168.1.111	C i 'B [[6*2.2S "7 E N 6#/ n · [* V II] e[* / OV II] hu?
1. 161.977999 192.168.1.108 192.168.1.108 102.168.1.111 192.168.1.111<	1.	. 161.977985	192.168.1.111	192.168.1.108	
1. 161.978094 192.168.1.111 192.168.1.108 TC 1. 161.978127 192.168.1.108 192.168.1.108 TC 1. 161.978135 192.168.1.108 192.168.1.108 TC 1. 161.978141 192.168.1.111 192.168.1.111 192.168.1.111 192.168.1.111 1. 161.978145 192.168.1.111 192.168.1.111 192.168.1.111 192.168.1.111 1. 161.978150 192.168.1.111 192.168.1.111 192.168.1.111 192.168.1.111 9. Options: (12 bytes), No-Operation (NOP), No-Operation (NOP) No-Operation (NOP) No-Operation (NOP) > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP) No-Operation (NOP) No-Operation (NOP) SEQ/ACK analysis	1.	. 161.977999	192.168.1.108	192.168.1.111	· · · · · · · · · · · · · · · · · · ·
1. 161.978127 192.168.1.108 192.168.1.101 TC 1. 161.978135 192.168.1.108 192.168.1.101 TC 1. 161.978135 192.168.1.101 192.168.1.101 TC 1. 161.978145 192.168.1.101 192.168.1.101 TC 1. 161.978145 192.168.1.101 192.168.1.101 TC 1. 161.978150 192.168.1.101 192.168.1.101 TC 1. 161.978150 192.168.1.101 HT HA.A.A.A.A.A.A.A.A.A.A.A.A.A.A.A.A.A.A.	1.	. 161.978004	192.168.1.111	192.168.1.108	· · · · · · · · · · · · · · · · · · ·
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	1	. 161.978127	192.168.1.108	192.168.1.111	c aaaaaaaaaaaaa
1161.978141 192.168.1.111 192.168.1.111 192.168.1.111 1161.978145 192.168.1.111 192.168.1.111 192.168.1.111 1161.978150 192.168.1.111 192.168.1.111 192.168.1.111 1161.978150 192.168.1.111 192.168.1.111 192.168.1.111 1161.978150 192.168.1.111 192.168.1.111 192.168.1.111 1161.978150 192.168.1.111 192.168.1.111 192.168.1.111 1161.978150 192.168.1.111 192.168.1.111 192.168.1.111 1161.978150 192.168.1.111 192.168.1.111 192.168.1.111 1161.978150 192.168.1.111 192.168.1.111 192.168.1.111 1161.978150 192.168.1.111 192.168.1.111 192.168.1.111 1161.978150 192.168.1.111 192.168.1.111 192.168.1.111 1161.978150 112.01.01.01 100.168.1.111 102.168.1.111 1161.978150 112.01.01.01 100.168.1.111 100.168.1.111 100.168.1.111 1161.978150 112.01.01.01 100.168.1.111 100.168.1.111 100.168.1.111 100.168.1.111 110.10.11 <td< td=""><td>1</td><td>. 161.978135</td><td>192.168.1.111</td><td>192.168.1.108</td><td></td></td<>	1	. 161.978135	192.168.1.111	192.168.1.108	
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	1	. 161.978141	192.168.1.108	192.168.1.111	c 0AAAAAAAAAAAA
1 161.978150 192.168.1.108 192.168.1.111 HT 1 161.978150 192.168.1.108 192.168.1.108 1400.100 1400.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1	1	. 161.978145	192.168.1.111	192.168.1.108	C
• Lot control • Lo	1.	. 161.978150	192.168.1.108	192.168.1.111	H +9AAAAAAAAAAAA.
<pre>> Options: (12 bytes), No-Operation (NOP), No-Operation (NOP) > [SEQ/ACK analysis] > [Timestamps] TCP payload (723 bytes) Retransmitted TCP segment data (723 bytes) Retransmitted TCP segment data (723 bytes) 0040 a8 36 50 4f 53 54 20 2f 63 6f 6e 66 31 67 2e 70 0050 68 70 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 0060 74 3a 20 31 39 32 2e 31 36 38 2e 31 2e 31 30 38 t: 192 0070 dd 0a 55 73 26 71 2e 31 6e 74 3a 20 4d 6f '-User 0060 74 3a 20 31 39 32 2e 31 36 38 2e 31 2e 31 30 38 t: 192 5V, V, V</pre>	1	161 070153	100 100 1 111	100 100 1 100	
<pre>> [SEQ/ACK analysis] > [Timestamps] TCP payload (723 bytes) Retransmitted TCP segment data (723 bytes) @040 a8 36 50 4f 53 54 20 2f 63 6f 6e 66 31 67 2e 70 0055 68 70 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 hp HTTF 0066 74 3a 20 31 39 32 2e 31 36 38 2e 31 2e 31 30 38 t: 192 0070 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f 'User 4: 0070 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f 'User 0080 3a 34 35 2e 30 29 20 47 65 6a 6b 6f 2f 32 30 31 :45.00 0060 3a 34 35 2e 30 29 20 47 65 6a 6b 6f 2f 32 20 31 :45.00 0060 3a 34 35 2e 30 29 20 47 65 6a 6b 6f 2f 32 20 31 :45.00 0060 3a 34 35 2e 30 29 20 47 65 6a 6b 6f 2f 32 20 31 :45.00 0060 3a 34 35 2e 30 29 20 47 65 6a 6b 6f 2f 32 20 31 :45.00 0060 3a 30 31 30 31 20 46 69 72 65 6a 6b 6f 2f 32 30 31 :45.00 0060 2f 68 74 6d 6c 2c 6f 70 70 6c 6f 78 2f 34 35 0010 f </pre>	>	Options: (12 bytes), N	Io-Operation (NOP),	No-Operation (NO	
<pre>> [Timestamps] TCP payload (723 bytes) Retransmitted TCP segment data (723 bytes) 0040 a8 36 50 4f 53 54 20 2f 63 6f 6e 66 31 67 2e 70</pre>	>	[SEQ/ACK analysis]			
TCP payload (723 bytes) Retransmitted TCP segment data (723 bytes) 0040 a8 36 50 4f 53 54 20 2f 63 6f 6e 66 31 67 2e 70 •6POST 0050 68 70 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 hp HTTF 0050 68 70 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 hp HTTF 0060 74 3a 20 31 39 32 2e 31 36 38 2e 31 2e 31 30 38 t: 192 0070 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f ·User- 1q	>	[Timestamps]			&
Retransmitted TCP segment data (723 bytes) 0040 a8 36 50 4f 53 54 20 2f 63 6f 6e 66 31 67 2e 70 •6POST 0050 68 70 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 hp HTTF 0060 74 3a 20 31 39 32 2e 31 36 38 2e 31 2e 31 30 38 t: 192 0080 7a 69 6c 6c 12 f3 52 2a 30 22 85 8 31 31 3b 20 zilla/5 0090 4c 69 6e 75 78 20 78 38 36 5f 36 34 3b 20 72 76 Linux x 0040 30 30 31 30 31 20 46 69 72 65 66 6f 78 2f 34 35 0010 f 0040 22 30 0d 0a 41 63 63 65 70 74 3a 20 74 65 78 74 0.0.46 57 78 40 72 76 66 78 2a 34 35 0.010 f 0040 30 63 13 00 21 20 66 67 74 3a 20 46 57 87 4 0.0.46 57 78 40 72 76 66 78 3a 20 31 145.00 7.n6.Y. 0040 26 86 74 66 6c 2 6 17 73 6c 6 74 3a 20 74 65 78 74 0.0.46 57 78 40 72 76 Linux x 0040 30 31 30 31 20 46 69 72 65 66 6f 78 2f 34 35 0010 f		TCP payload (723 bytes	5)		.@.J4:q.A,M5~tQ00%.JiEt\$2.q.(
0040 a8 36 50 4f 53 54 20 2f 63 6f 6e 66 31 67 2e 70 •6POST 0050 68 70 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 hp HTTF 0060 74 3a 20 31 39 32 2e 31 36 38 2e 31 2e 31 30 38 t: 192, 0070 00 a 65 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f		Retransmitted TCP segn	nent data (723 bytes)	APa
0040 a8 36 50 4f 53 54 20 2f 63 6f 6e 66 31 67 2e 70 -6P0ST		-			_ !q
0050 68 70 20 48 54 56 27 31 22 31 20 31 24 31 24 31 24 31 24 31 24 31 24 31 24 31 24 31 24 31 33 32 24 31 24 31 30 8 t: 192 \$VY^9,.*`a.M7.VXKa/K;W DSK.H.H/[W DSK.H.BK.f.H.B. 9000 42 35 57 32 24 31 30 8 t: 192 \$VY^9,.*`a.M7.VXKa/Ka/K'AK;W DSK.A.L/K.H.BKH.BKA'LKKa/KA'LKKa/KA'LKKa/	0040	a8 36 50 4f 53 54 20	2f 63 6f 6e 66 31	67 2e 70 •6P09	
0060 74 32 26 31 36 38 26 31 29 32 26 31 36 38 26 31 29 32 26 31 36 38 26 31 29 32 26 31 36 38 26 31 29 32 26 31 36 38 26 31 26 31 26 31 26 31 26 31 26 31 26 36 66 67 35 22 20 28 58 31 31 30 20 211a/5 4Fugs, +.u., 2.k., U.z.k., U.= +>>, 2, .u.keu. }ac.RP1+, y.e. *.[], .u.f.f.8 4Fugs, +.u., Z.k.Y.+>>, 2, .u.keu. }ac.RP1+, y.e. *.[], .u.f.f.8 4Fugs, +.u., Z.k.Y.+>>, 2, .u.keu. }ac.RP1+, y.e. *.[], .u.f.f.8 4Fugs, +.u.Z.k.Y.+>>, 2, .u.keu. }ac.K. .u.f.g.g.o.u.f.f.f.f.f.f.f.f.f.f.f.f.f.f.f.f.f.f	0050	68 70 20 48 54 54 50	2+ 31 2e 31 0d 0a	48 6† 73 hp H1	TF 3!F@KI"1@.+fcXE~".[LfG^.99ikmWT\[.z y*<
0070 00 04 05 05 06 74 34 20 40 05 05 06 74 34 20 40 05 05 06 74 34 20 40 05 05 06 74 34 20 40 05 05 06 74 34 20 40 05 05 06 74 34 20 40 05 05 06 74 34 20 06 74 34 20 06 74 34 20 06 75 75 20 78 38 36 57 36 54 54 20 72 1110.x , , , , , , , , , , , , ,	0060	74 3a 20 31 39 32 2e	31 36 38 2e 31 2e	31 30 38 T: 19	2 \$V\Y^.9*. a.M7.VxKa/%;;W DsF.A!=/+0B1q;9YtR
0080 74 09 06 <	0070		41 67 65 66 74 3a	20 40 6T ··Use	Γ #.J4Jt.Dt.DfB
0030 3a 35 20 7a 3a 3b 3a 3b 20 4a 6b 7a 5a 6a 6a 2a 3a 3a 3a 3a 2a 2a 2a 3a 3a 2a 7a 5a 6a 6a 7a 7a 5a 6a 6a 7a 7a 5a 6a 7a 7a 5a 6a 7a 7a <	0080		28 30 20 28 38 31	31 30 20 ZIIIa	4FD#3^.+V.Z.k.Y+>>.).2uk{k.h,1.0.3I&g.,01pY.
00b0 30 31 30 31 20 46 69 72 65 66 67 82 53 00101 [] []	0090	3a 34 35 2e 30 29 20	47 65 63 6h 6f 2f	32 30 31 · 45 0	1 J D.myQE9 pK
0000 2e 30 00 a 12 63 65 70 74 3a 20 74 65 78 74 57 74 3a 20 74 65 78 74 58 74 66 26 74 64 62 62 61 70 74 63 61 74 69 61 /html,a /html,a <td>aaha</td> <td>30 30 31 30 31 20 46</td> <td>69 72 65 66 6f 78</td> <td>2f 34 35 00101</td> <td>$F = 0$ * by γp_{C} is by C^{γ} stability</td>	aaha	30 30 31 30 31 20 46	69 72 65 66 6f 78	2f 34 35 00101	$F = 0$ * by γp_{C} is by C^{γ} stability
00d0 2f 68 74 6d 6c 2c 61 70 70 6c 69 63 61 74 69 6f //html, · 00e0 6e 2f 78 68 74 6d 6c 2b 78 6d 6c 2c 61 70 70 6c n/xhtml, · · 00f0 69 63 61 74 69 6f 6e 2f 78 6d 6c 2b 61 70 70 6c n/xhtml, · · · · 00f0 69 63 61 74 69 6f 6e 2f 78 6d 6c 3b 71 3d 30 2e ication ication ication ·	0000	2e 30 0d 0a 41 63 63	65 70 74 3a 20 74	65 78 74 .0	C =
00e0 6e 2f 78 68 74 6d 6c 2b 78 6d 6c 2c 61 70 70 6c n/xhtml 00f0 69 63 61 74 69 6f 6e 2f 78 6d 6c 3b 71 3d 30 2e ication 0100 39 2c 2a 2f 2a 3b 71 3d 30 2e 38 0d 0a 41 63 63 9,*/*;c 0110 65 70 74 2d 4c 61 6e 67 75 61 67 65 3a 20 65 6e ept-Lar	00d0	2f 68 74 6d 6c 2c 61	70 70 6c 69 63 61	74 69 6f /htm]	se (N`k.llullT.D
00f0 69 63 61 74 69 66 2f 78 6d c 3b 71 3d 30 2e ication 6i.*.N.LSq; Nicfl4g:{ftop_Is_Waiting_4_y} 6i.*. 0100 39 2c 2a 2f 2a 3b 71 3d 30 2e 38 0d 0a 41 63 39 9,*/*;c ept-Lar /// # 13330.1 #/# # /# /# /# /# /# /# /# /# /# /# /# /	00e0	6e 2f 78 68 74 6d 6c	2b 78 6d 6c 2c 61	70 70 6c n/xht	ml ?.1.b9.;n5f]\$.6r0.bD*.WMa.B+a <bh< td=""></bh<>
0100 39 2c 2a 2f 2a 3b 71 3d 2e 38 0d a 41 63 63 9,*/*;c	00f0	69 63 61 74 69 6f 6e	2f 78 6d 6c 3b 71	3d 30 2e icati	or 6i.*.NLsq;.Nicfl4g:{ftop Is Waiting 4 y}
0110 65 70 74 2d 4c 61 6e 67 75 61 67 65 3a 20 65 6e ept-Lar 州细 13330.1 卷户碑 州细,1 液分晶 州纲,1 rarn(s) 点曲选择。	0100	39 2c 2a 2f 2a 3b 71	3d 30 2e 38 0d 0a	41 63 63 9,*/*	;q
	0110	65 70 74 2d 4c 61 6e	67 75 61 67 65 3a	20 65 6e ept-L	an 分组 13330。1 <u>客户端</u> 分组,1 服务据 分组,1 turn(s). 点击选择。

42.很普通的数独