2019红帽杯(RedHat)【Broadcastc & 恶臭的数据包】writeup

原创

「已注销」 于 2019-11-11 20:17:31 发布 2406 收藏 6
文章标签： 2019 红帽杯 CTF 数据分析 无线

首先请原谅我这么菜，只解了两道题还来发Blog；

【Broadcastc】

这题不多说，直接解压得到task.py打开获取flag

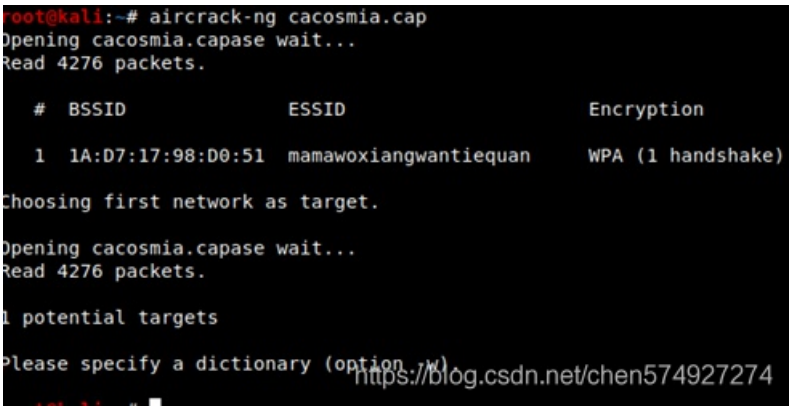
【恶臭的数据包】

首先下载压缩包解压得到数据包

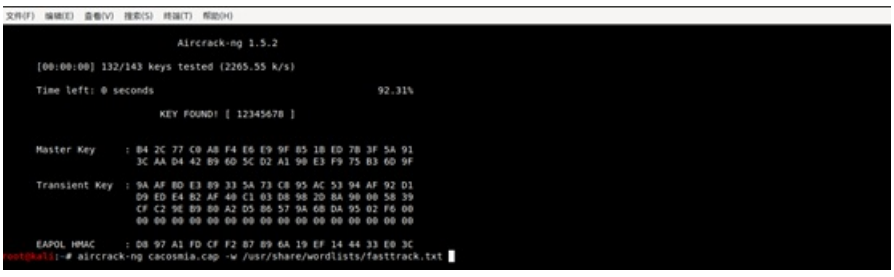
Wireshark打开什么都看不懂，直接百度；

按照教程　把数据包丢到Kali

先看看ESSID

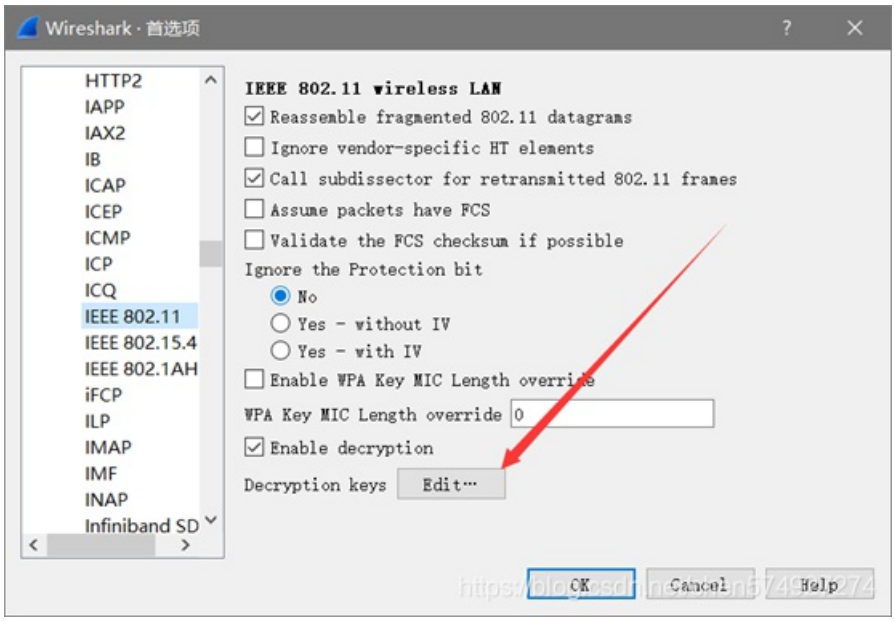


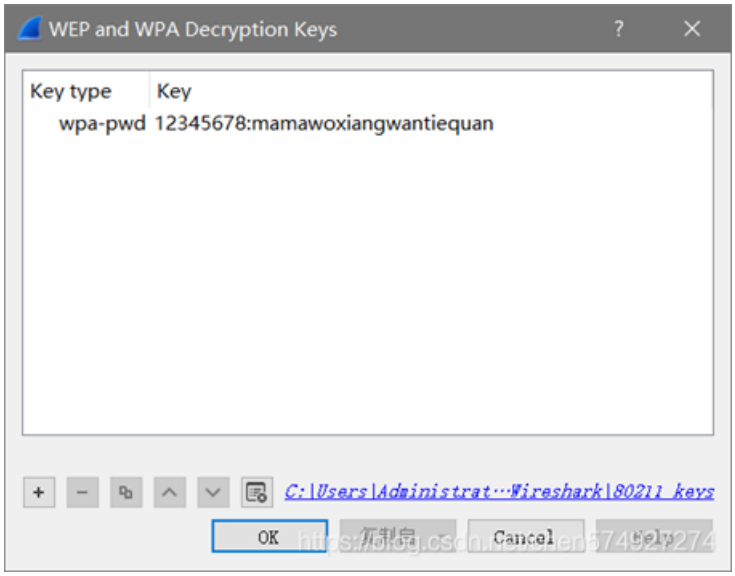
然后再破解WiFi密码



得到WiFi密码 12345678

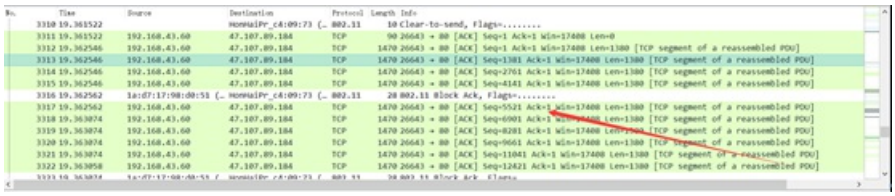
打开Wireshark 编辑->首选项->Protocols->IEEE 802.11



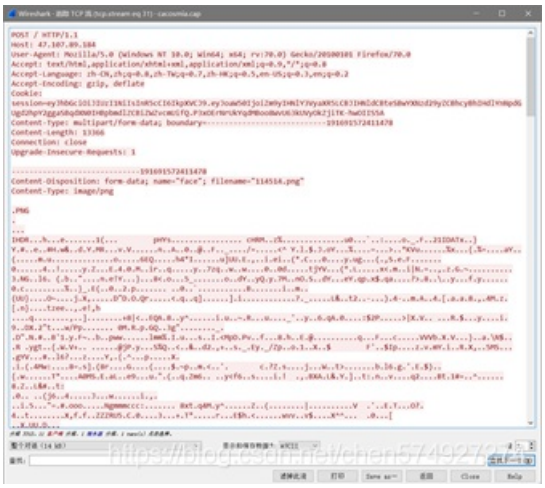


设置wpa-pwd 值的格式(密码:WiFi名称/pass:essid) 保存之后发现数据包被解密了

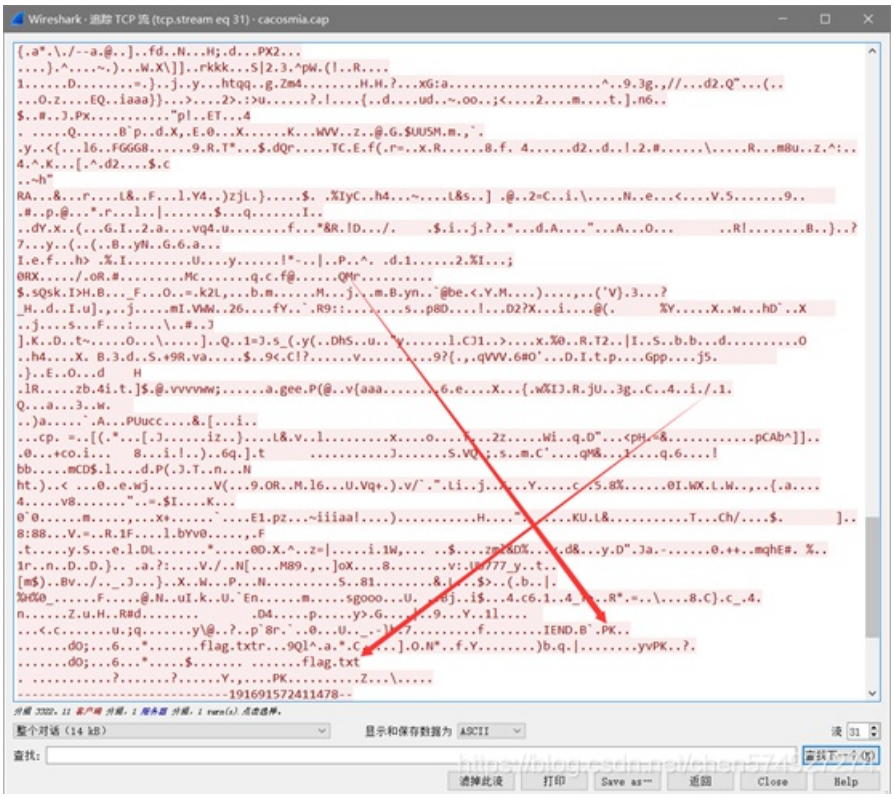
通过分析发现，有个可疑的HTTP请求



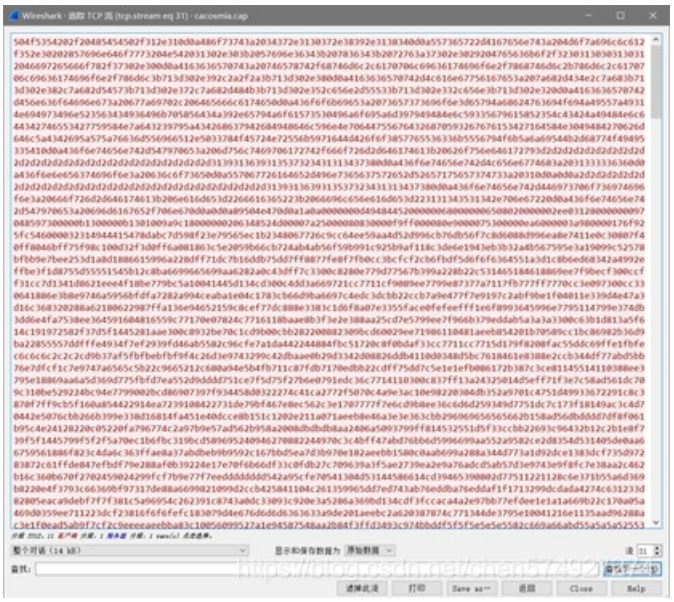
右击 追踪流->TCP



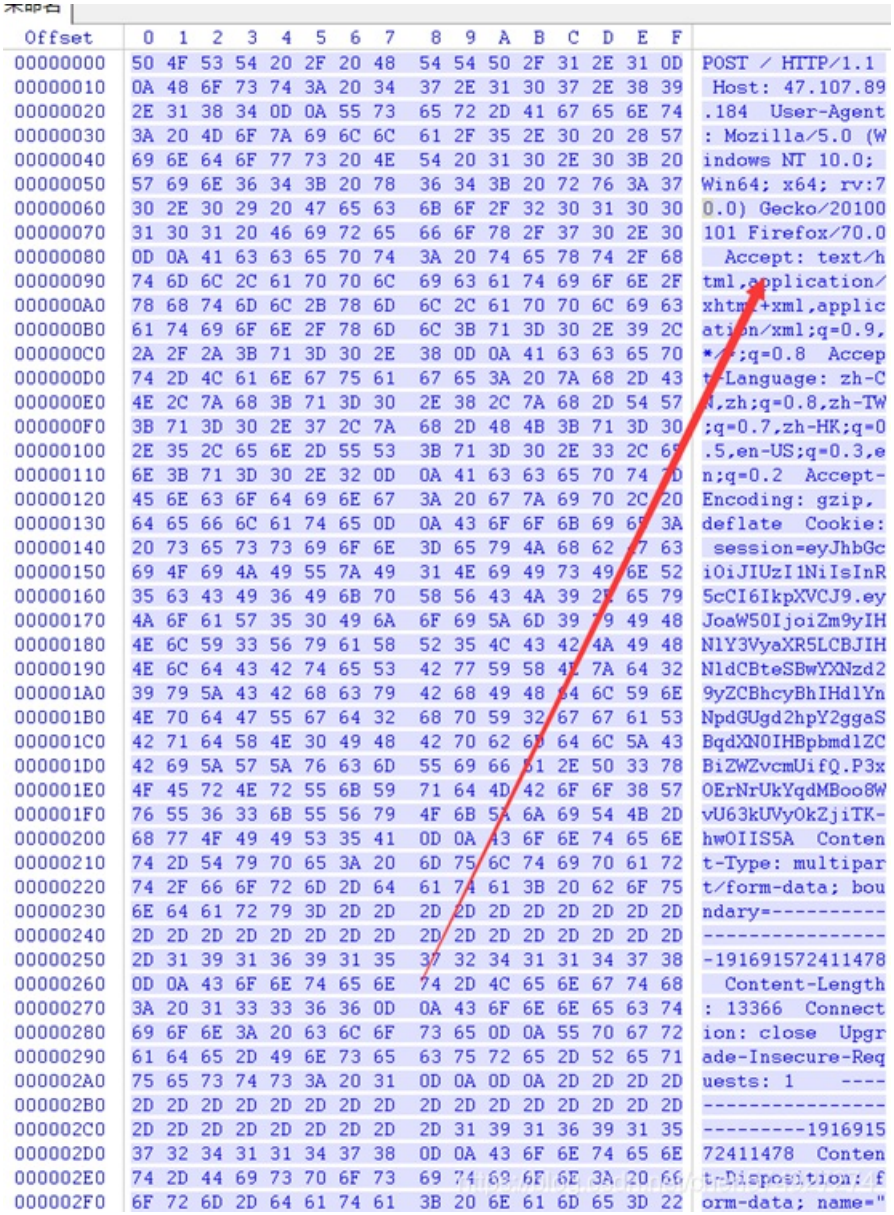
发现是个文件上传，末尾发现还有一个ZIP。



将　显示和保存数据为　改成原始数据



复制进winhex十六进制编辑器　新建文件->粘贴->选择 ASCII HEX

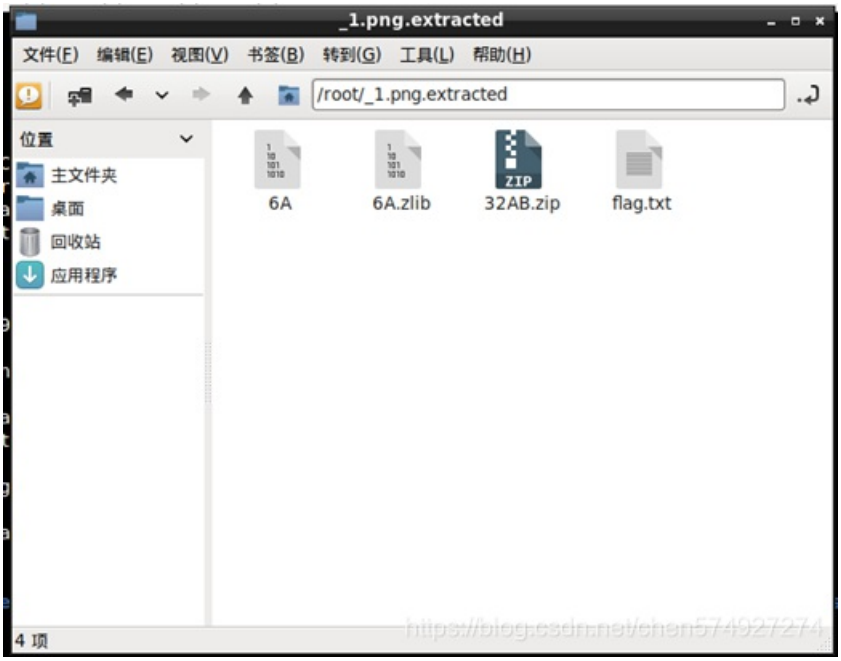


Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	50	4F	53	54	20	2F	20	48	54	54	50	2F	31	2E	31	0D	POST / HTTP/1.1
00000010	0A	48	6F	73	74	3A	20	34	37	2E	31	30	37	2E	38	39	Host: 47.107.89
00000020	2E	31	38	34	0D	0A	55	73	65	72	2D	41	67	65	6E	74	.184 User-Agent
00000030	3A	20	4D	6F	7A	69	6C	6C	61	2F	35	2E	30	20	28	57	: Mozilla/5.0 (W
00000040	69	6E	64	6F	77	73	20	4E	54	20	31	30	2E	30	3B	20	indows NT 10.0;
00000050	57	69	6E	36	34	3B	20	78	36	34	3B	20	72	76	3A	37	Win64; x64; rv:7
00000060	30	2E	30	29	20	47	65	63	6B	6F	2F	32	30	31	30	30	0.0) Gecko/20100
00000070	31	30	31	20	46	69	72	65	66	6F	78	2F	37	30	2E	30	101 Firefox/70.0
00000080	0D	0A	41	63	63	65	70	74	3A	20	74	65	78	74	2F	68	Accept: text/h
00000090	74	6D	6C	2C	61	70	70	6C	69	63	61	74	69	6F	6E	2F	tml,application/
000000A0	78	68	74	6D	6C	2B	78	6D	6C	2C	61	70	70	6C	69	63	xhtml+xml,applic
000000B0	61	74	69	6F	6E	2F	78	6D	6C	3B	71	3D	30	2E	39	2C	ation/xml;q=0.9,
000000C0	2A	2F	2A	3B	71	3D	30	2E	38	0D	0A	41	63	63	65	70	*/*;q=0.8 Accep
000000D0	74	2D	4C	61	6E	67	75	61	67	65	3A	20	7A	68	2D	43	t-Language: zh-C
000000E0	4E	2C	7A	68	3B	71	3D	30	2E	38	2C	7A	68	2D	54	57	N,zh;q=0.8,zh-TW
000000F0	3B	71	3D	30	2E	37	2C	7A	68	2D	48	4B	3B	71	3D	30	;q=0.7,zh-HK;q=0
00000100	2E	35	2C	65	6E	2D	55	53	3B	71	3D	30	2E	33	2C	65	.5,en-US;q=0.3,e
00000110	6E	3B	71	3D	30	2E	32	0D	0A	41	63	63	65	70	74	2D	n;q=0.2 Accept-
00000120	45	6E	63	6F	64	69	6E	67	3A	20	67	7A	69	70	2C	20	Encoding: gzip,
00000130	64	65	66	6C	61	74	65	0D	0A	43	6F	6F	6B	69	65	3A	deflate Cookie:
00000140	20	73	65	73	73	69	6F	6E	3D	65	79	4A	68	62	47	63	session=eyJhbGc
00000150	69	4F	69	4A	49	55	7A	49	31	4E	69	49	73	49	6E	52	iOiJIUzI1NiIsInR
00000160	35	63	43	49	36	49	6B	70	58	56	43	4A	39	2E	65	79	5cCI6IkpXVCJ9.ey
00000170	4A	6F	61	57	35	30	49	6A	6F	69	5A	6D	39	79	49	48	JoaW50IjoiZm9yIH
00000180	4E	6C	59	33	56	79	61	58	52	35	4C	43	42	4A	49	48	NlY3VyaXR5LCBJIH
00000190	4E	6C	64	43	42	74	65	53	42	77	59	58	4E	7A	64	32	NldCBteSBwYXNzd2
000001A0	39	79	5A	43	42	68	63	79	42	68	49	48	64	6C	59	6E	9yZCBhcyBhIHdlYn
000001B0	4E	70	64	47	55	67	64	32	68	70	59	32	67	67	61	53	NpdGUgd2hpY2ggaS
000001C0	42	71	64	58	4E	30	49	48	42	70	62	6D	64	6C	5A	43	BqdXN0IHBpbmdlZC
000001D0	42	69	5A	57	5A	76	63	6D	55	69	66	51	2E	50	33	78	BiZWZvcmUifQ.P3x
000001E0	4F	45	72	4E	72	55	6B	59	71	64	4D	42	6F	6F	38	57	OErNrUkYqdMBoo8W
000001F0	76	55	36	33	6B	55	56	79	4F	6B	5A	6A	69	54	4B	2D	vU63kUVyOkZjiTK-
00000200	68	77	4F	49	49	53	35	41	0D	0A	43	6F	6E	74	65	6E	hwOIIS5A Conten
00000210	74	2D	54	79	70	65	3A	20	6D	75	6C	74	69	70	61	72	t-Type: multipar
00000220	74	2F	66	6F	72	6D	2D	64	61	74	61	3B	20	62	6F	75	t/form-data; bou
00000230	6E	64	61	72	79	3D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	ndary----------
00000240	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	----------
00000250	2D	31	39	31	36	39	31	35	37	32	34	31	31	34	37	38	-191691572411478
00000260	0D	0A	43	6F	6E	74	65	6E	74	2D	4C	65	6E	67	74	68	Content-Length
00000270	3A	20	31	33	33	36	36	0D	0A	43	6F	6E	6E	65	63	74	: 13366 Connect
00000280	69	6F	6E	3A	20	63	6C	6F	73	65	0D	0A	55	70	67	72	ion: close Upgr
00000290	61	64	65	2D	49	6E	73	65	63	75	72	65	2D	52	65	71	ade-Insecure-Req
000002A0	75	65	73	74	73	3A	20	31	0D	0A	0D	0A	2D	2D	2D	2D	uests: 1 ----
000002B0	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	2D	----------
000002C0	2D	2D	2D	2D	2D	2D	2D	2D	2D	31	39	31	36	39	31	35	---------1916915
000002D0	37	32	34	31	31	34	37	38	0D	0A	43	6F	6E	74	65	6E	72411478 Conten
000002E0	74	2D	44	69	73	70	6F	73	69	74	69	6F	6E	3A	20	66	t-Disposition: f
000002F0	6F	72	6D	2D	64	61	74	61	3B	20	6E	61	6D	65	3D	22	orm-data; name="

将前面的HTTP头部信息去除，修复PNG图片，之后保存得到一张并没什么用的图片

将图片丢到Kali　binwalk -e xx.png

将zip分解出来，可能是文件尾部清理不干净，导致binwalk卡住了，回车一下(也可以尝试修复一下)



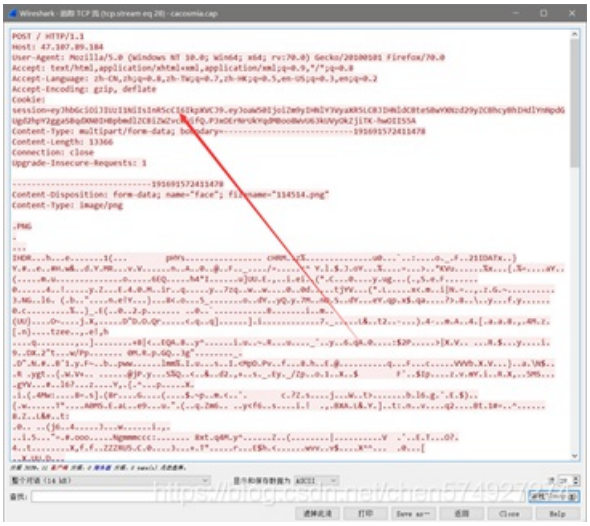
得到以上文件夹，别以为看到flag就以为自己成功了

打开为空，哈哈哈哈

将zip拷贝到windows　放到archpr跑包，一直未果。

这就纳闷了。。。。。

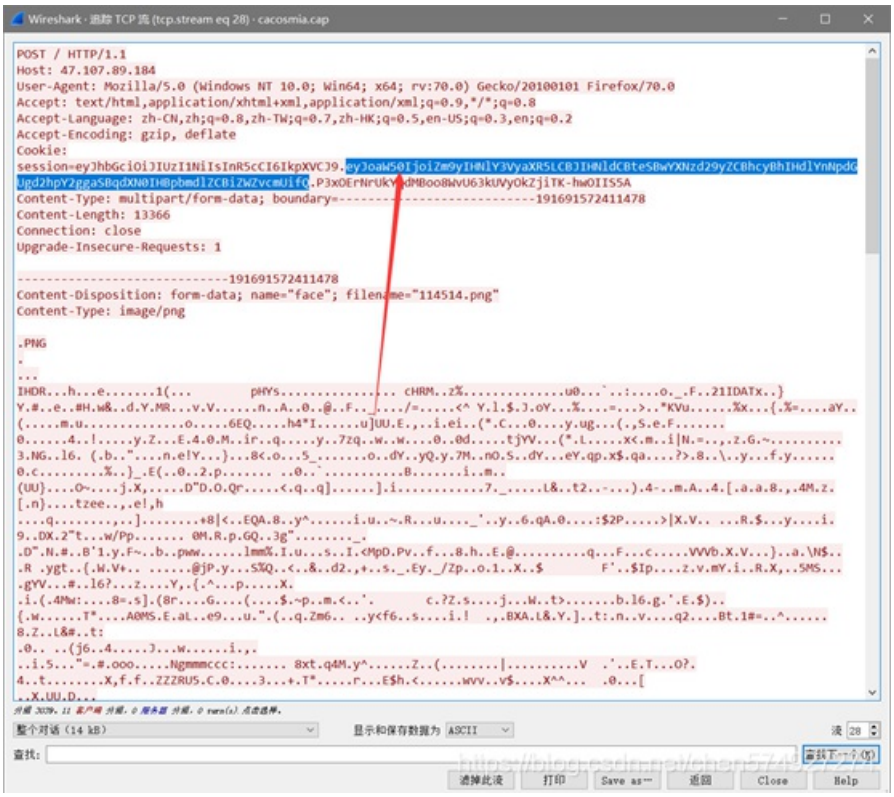
后来分析发现请求头部的Cookie



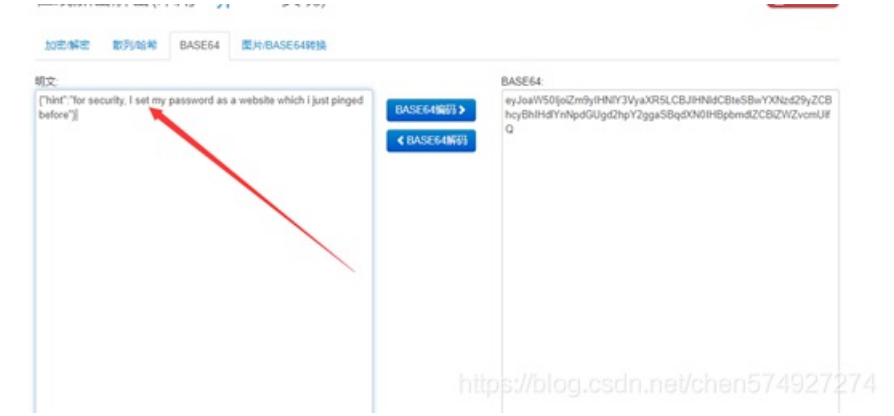
后来得知是jwt

不懂可以看看　https://juejin.im/post/5a437441f265da43294e54c3

按照网上原理得知是base64加密，拷贝中间的payload



通过base64解密，得到如下

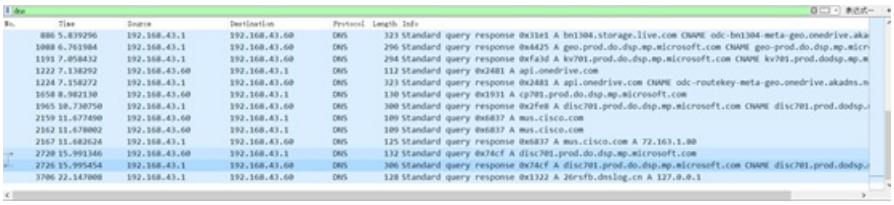


请原谅英语差的我

Ping 无非就是ICMP协议；Wireshark过滤ICMP

发现没有任何东西

后来灵感一来，网站域名解析肯定就是DNS了；



通过刚刚的HTTP请求的序号，这里的DNS肯定就是后面点的

试了好几个，最后发现是最后一个域名

26rsfb.dnslog.cn

把域名当密码解压ZIP

得到flag