掘安平台Writeup(持续解题)

notwiner ① 于 2020-01-29 22:34:04 发布 ② 821 🏫 收藏 4 分类专栏: CTF 文章标签: 密码学 版权声明:本文为博主原创文章,遵循<u>CC 4.0 BY-SA</u>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/gg 41668936/article/details/104110985 版权

CTF 专栏收录该内容

4篇文章1订阅

订阅专栏

掘安平台Writeup

刷一部分sqli-labs,先掘安平台做点CTF,主要目的是练习web,MSIC和crypto则顺便。

misc

welcome

没什么说的,关注公众号jasafe110然后发flag就行。

hello

流量分析基础题目,工具wireshark

导入后为下图界面

hello	o.pcapng				
文件(F)	编辑(E) 视图(V) 题	W转(G) 捕获(C) 分析(A) 纷	钻计(S) 电话(Y) 无线(W)	工具(T) 帮助(H	4)
(=	1 🔘 📙 🖻 🗙 🕻	🕽 ९ 🗢 🗢 🕾 💽 🛓	📃 @, @, @, 🎹		
、应用	显示过滤器 … ‹Ctrl-/>				
lo.	Time	Source	Destination	Protocol L	Info
	1 0.000000	192.168.100.238	180.97.33.107	TCP	54 1399 → 443 [FIN, ACK] Seq=1 Ack=1 Win=1024 Len=0
	2 0.647187	192.168.100.238	180.97.33.107	TCP	66 1421 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
	3 0.681082	180.97.33.107	192.168.100.238	TCP	66 443 → 1421 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1452 WS=32 SACK_PERM=1
	4 0.681147	192.168.100.238	180.97.33.107	TCP	54 1421 → 443 [ACK] Seq=1 Ack=1 Win=262144 Len=0
	5 0.681292	192.168.100.238	180.97.33.107	TCP	54 1421 → 443 [FIN, ACK] Seq=1 Ack=1 Win=262144 Len=0
	6 0.813737	192.168.100.227	192.168.100.255	UDP	305 54915 → 54915 Len=263
	7 0.982253	192.168.100.238	180.97.33.107	TCP	54 [TCP Retransmission] 1421 → 443 [FIN, ACK] Seq=1 Ack=1 Win=262144 Len=0
	8 1.521476	192.168.100.238	123.151.78.53	QICQ	81 OICQ Protocol
	9 1.569905	123.151.78.53	192.168.100.238	OICQ	89 OICQ Protocol
	10 1.583991	192.168.100.238	180.97.33.107	тср	54 [TCP Retransmission] 1421 → 443 [FIN, ACK] Seq=1 Ack=1 Win=262144 Len=0
	11 1.837745	192.168.100.227	192.168.100.255	UDP	305 54915 → 54915 Len=263
	12 2.783946	192.168.100.238	180.97.33.107	ТСР	54 [TCP Retransmission] 1421 → 443 [FIN, ACK] Seq=1 Ack=1 Win=262144 Len=0
	13 2.862130	192.168.100.227	192.168.100.255	UDP	305 54915 → 54915 Len=263
	14 3.783755	192.168.100.227	192.168.100.255	UDP	305 54915 → 54915 Len=263
	15 3.811703	fe80::c59:47a9:8479	ff02::2	ICMPv6	70 Router Solicitation from 88:ae:07:d1:9d:85
	16 4.165246	192.168.100.195	224.0.0.251	MDNS	112 Standard query 0x0000 PTR _sleep-proxyudp.local, "QU" question OPT
	17 4.166476	fe80::c59:47a9:8479	ff02::fb	MDNS	132 Standard query 0x0000 PTR _sleep-proxyudp.local, "QU" question OPT
	18 / 211051	£080	ff0Jth	TCMDVG	86 Multicast Listanan Ranant
> Fra	me 37: 70 bytes o	on wire (560 bits), 70	bytes captured (560	bits) on i	nterface 0
N Eth	onnot TT Spc. 99		0.07.d1.0d.95) Det.	Arupourou 7	9-54-47 (89-55-89-79-54-47)

> Internet II, STC: 80:30:01/01/90:50 (80:30:07/01/90:50), DST: AZUPEWWWY/S > Internet Protocol Version 6, STC: fe80::c59:47a9:8479:40b7, DST: ff02::2 > Internet Control Message Protocol v6

0000	80	a5	89	79	54	d7	88	ae	07	d1	9d	85	86	dd	60	Ød	yT	
0010	61	70	00	10	3a	ff	fe	80	00	00	00	00	00	00	0c	59	ap:	Y
0020	47	a9	84	79	40	b7	ff	02	00	00	00	00	00	00	00	00	Gy@	
0000	~~	~~	~~	~~	~~	00	or	00	24	**	~~	-	-		04	04		

点工具栏的Protocol使之通过传输协议排序,或者直接通过上面的过滤器,找TCP和HTTP即可(基础类型的一般这两个够用)。

挨个点,注意最下面的窗口会显示传输具体内容,发现有一个time为11.930535的,编号为73的有flag字样。但是这个现在没办 法复制,选中有flag字样的那一行后右键——追踪流——TCP或者HTTP均可,然后现在就可以复制了。

```
POST /config.inc.php HTTP/1.1
Host: 192.168.100.200:8081
Accept-Encoding: gzip, deflate
User-Agent: antSword/1.1
Content-Type: application/x-www-form-urlencoded
Content-Length: 342
Connection: close
08067sec=%40ini_set(%22display_errors%22%2C%20%220%22)%3B%40set_time_limit(0)%3Becho%20%22-%3E%7C%22%3B%24F%3Dbase64 decode(%24 POST
%220xbcedd6b0aae69%22%5D)%3B%24P%3D%40fopen(%24F%2C%22r%22)%3Becho(%40fread(%24P%2Cfilesize(%24F)))%3B%40fclose(%24P)%3B%3Becho%20%2
%3B&0xbcedd6b0aae69=L3Zhci93d3cvaHRtbC9mbGFnLnBocA%3D%3DHTTP/1.1 200 OK
Date: Tue, 11 Dec 2018 14:02:33 GMT
Server: Apache/2.4.29 (Ubuntu)
Content-Length: 57
Connection: close
Content-Type: text/html; charset=UTF-8
-> <?php
    $flag = 'SWPUCTF{Th1s_i3_e4sy_pc@p}';
?>
<-
```

https://blog.csdn.net/qq_41668936

tips: 流量分析这些即为平时使用浏览器访问所产生的数据流,有不同协议类型(TCP、HTTP、UDP等等),具体可以找 wireshark或者流量分析的书看,之前初始wireshark有过部分总结。

misc-hello

解压是个图片,字节分析HxD看一眼,搜索下flag,结束

) 文件(F) 编	辑(E)	搜	素(S)) 视	图(V) 分	新(A	L (具(1) 種		W)	帮助	(H)			
) 👌 - 🐻				•	+ +	16		~ 1	Wind	ows	(AN	ISI)		~	+>	进制	
Easy.png																	
)ffset(h)	00	01	02	03	04	05	06	07	08	09	OA	0B	0C	0D	0E	OF	对应文本
00040F20	7E	54	E6	51	14	45	51	14	C5	F8	51	99	47	51	14	45	~TæQ.EQ.ÅøQ™GQ.E
00040F30	51	14	E3	47	65	1E	45	51	14	45	51	8C	1F	95	79	14	Q.ãGe.EQ.EQE.•y.
00040F40	45	51	14	45	31	7E	54	E6	51	14	45	51	14	C5	F8	51	EQ.E1~TæQ.EQ.ÅøQ
00040F50	99	47	51	14	45	51	14	E3	47	65	1E	45	51	14	45	51	™GQ.EQ.ãGe.EQ.EQ
00040F60	8C	lF	95	79	14	45	51	14	45	31	7E	54	E6	51	14	45	C. • y.EQ.E1~TæQ.E
00040F70	51	14	C5	F8	51	99	47	51	14	45	51	14	E3	47	65	1E	Q.ÅøQ™GQ.EQ.ãGe.
00040F80	45	51	14	45	51	8C	1F	95	79	14	45	51	14	45	31	7E	EQ.EQC. • y.EQ.E1~
00040F90	54	E6	51	14	45	51	14	C5	F8	51	99	47	51	14	45	51	TæQ.EQ.ÅøQ™GQ.EQ
00040FA0	14	E3	47	65	1E	45	51	14	45	51	8C	lF	95	79	14	45	.ãGe.EQ.EQC.•y.E
00040FB0	51	14	45	31	7E	54	E6	51	14	45	51	14	C5	F8	51	99	Q.E1~TæQ.EQ.ÅøQ™
00040FC0	47	51	14	45	51	14	E3	47	65	1E	45	51	14	45	51	8C	GQ.EQ.ãGe.EQ.EQE
00040FD0	17	8F	7B	DC	FF	07	52	06	FD	52	D6	12	85	BO	00	00	{Üÿ.R.ýRÖ°
00040FE0	00	1B	74	45	58	74	41	72	74	69	73	74	00	66	6C	61	tEXtArtist.fla
00040FF0	67	7B	62	32	62	38	35	65	63	37	65	63	38	63	63	34	g{b2b85ec7ec8cc4
00041000	37	A5	BB	FO	0C	00	00	00	00	49	45	4E	44	AE	42	60	7¥≫ðIEND⊗B`
00041010	82																,

https://blog.csdn.net/gg_41668936

disk

磁盘里藏着flag

LIVE TURNIT ARE LEADING

解压后发现两个文件,flag.dmg和.DS_Store,明显flag.dmg有用,放HxD看一下,

大多为乱码,搜索flag又没有有价值的线索,滑动着看,发现临近底部时候出现了%PNG,图片png标识也是这个,复制从%PNG的十六进制数,直到00结尾前的IEND®B`,这个也是PNG的结束位,新建粘贴另存为xx.png,发现flag

flag{m0unt_im4g3}

图片对比

两张图片有什么不一样吗?

解压,发现两个图片,flag.png和xor.png,放到HxD发现前面部分只有开头有差别,联系上题目为图片对比,猜测异或。(其实 也可能是先修复文件头然后放到stegsolve里面对比,但是这里是字符异或)

上脚本,第一次做题写python脚本,不太熟练。(大佬们轻点吐槽)

```
ab = open("flag.png", "rb").read(20)
print(ab)
ac = open("xor.png","rb").read(20)
print(ac)
ab_1 = list(ab);
ac_1 = list(ac);
print(ab_1)
print(ac_1)
ad = ['x' for n in range(20)]
for i in range(19) :
    ad[i] = ab_1[i] ^ ac_1[i]
    i = i + 1
print (ad)
for i in range(19):
    ad[i] = chr(ad[i])
   i = i + 1
ad = "".join(str(i) for i in ad)
print(ad)
```

然后出现结果

mg", "rb"). 🎯 Python 3.7.4 Shell Х _ File Edit Shell Debug Options Window Help Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32 Type "help", "copyright", "credits" or "license()" for more information. in range(2 >>> .9) : | ^ ac_1 ------RESTART: A:\临时\题目\神奇的图片\1.py =------==== [i] ----b'\xef</ vct~3r5y &#*orx>'
b'\xef</ vct~3r5y &#*orx>'
b'\x89PNG\r\n\x1a\n\x00\x00\rIHDR\x00\x00\x05>'
[239, 60, 47, 32, 118, 99, 116, 126, 51, 114, 53, 121, 32, 38, 35, 42, 111, 114, 120, 62]
[137, 80, 78, 71, 13, 10, 26, 10, 0, 0, 0, 13, 73, 72, 68, 82, 0, 0, 5, 62]
[102, 108, 97, 103, 123, 105, 110, 116, 51, 114, 53, 116, 105, 110, 103, 120, 11
1, 114, 125, 'x']
[1ag/ist2rEfingurary .9): (ad[i]) :(i) for i flag{int3r5tingxor}x

颜表情

小姐姐

漂亮的小姐姐



(非上图,原图无法显示,截屏的)一看,图片尺寸不太对劲,有点太宽太低了,猜测是修改像素值,先看原始尺寸为1566*798,十进制转十六进制,得到061e*031e,放到HxD里面搜索这两个,两个在临近才对,随便修改下为041e,通关。



flag(H1gh_4nD_Wid7h)

https://blog.csdn.net/qq_41668936

真正的黑客才可以看到本质

真正的黑客才可以看到本质

解压,hacker.png,放到StegSolve,看各个色位的效果,Blue plane 0出现二维码,扫码或者QR识别,得到flag。

隐藏在黑夜里的秘密

隐藏在黑夜里的秘密

解压发现加密,放到zip密码爆破工具里面发现异常,应该是伪加密,

放到HxD里面,搜索504B(即文件头),找1400后面的,这里面是0008或者0908,挨个修改09为00(记得文件右键解除锁定),有两个需要修改,(下图为修改后的部分)

) 👌 🕶 🔚		CH .	đ	•	+ +	16	`	- 1	Nind	ows	(AN	ISI)		~	+>	进制	~
) yincangzai	heiye	elide	mim	i.zip													
Offset(h)	00	01	02	03	04	05	06	07	08	09	OA	0B	0C	OD	0E	OF	对应文本
00026000	04	C4	35	03	81	DO	59	57	53	OF	34	75	73	0B	5C	C3	.Ä5ĐYWS.4us.\Ã
00026010	22	04	3E	Al	B8	B3	B3	BD	8D	DF	D3	D7	D5	21	EA	EA	".>;, "35.BÓ×Õ!êê
00026020	68	69	6B	EC	18	98	9D	1A	1A	5E	79	F9	7C	7A	7E	65	hikì.~^yù z~e
00026030	FD	D9	9B	OF	5F	DF	3C	90	1F	99	98	98	9A	1A	1E	9B	ýÙ>. ß<œ.™~~š>
00026040	85	5D	EE	15	BD	92	AE	EE	6E	7E	A7	44	DO	D6	DC	58]î.½′®în~§DĐÖÜX
00026050	DF	98	2F	90	76	8B	14	12	41	4B	43	53	63	79	75	49	ß~/.v <akcscyui< td=""></akcscyui<>
00026060	41	72	FC	F5	A4	D4	C4	94	94	F8	D8	FF	1F	50	4B	01	Arüõ¤ÔÄ""øØÿ.PK.
00026070	02	3F	00	14	00	00	08	08	00	4F	AA	5E	4B	47	OF	28	.?Oª^KG.(
00026080	40	57	00	00	00	52	00	00	00	08	00	24	00	00	00	00	@W\$\$
00026090	00	00	00	20	00	00	00	00	00	00	00	66	6C	61	67	2E	flag.
000260A0	74	78	74	AO	00	20	00	00	00	00	00	01	00	18	00	14	txt
000260B0	Fl	5B	93	81	51	D3	01	81	CF	20	7D	80	51	D3	01	81	ñ[".QÓÏ }€QÓ
000260C0	CF	20	7D	80	51	D3	01	50	4B	01	02	3F	00	14	00	00	Ï }€QÓ.PK?
000260D0	08	08	00	0E	A9	5E	4B	45	BE	D5	A8	C3	5F	02	00	F6	©^KE¾Õ¨Ãö
000260E0	F2	05	00	OF	00	24	00	00	00	00	00	00	00	20	00	00	ò\$
000260F0	00	7D	00	00	00	54	72	65	65	69	6E	62	6C	61	63	6B	.}Treeinblack
00026100	2E	62	6D	70	OA	00	20	00	00	00	00	00	01	00	18	00	.bmp
00026110	29	00	CE	2C	80	51	D3	01	29	00	CE	2C	80	51	D3	01).Î,€QÓ.).Î,€QÓ.
00026120	63	91	76	FA	7E	51	D3	01	50	4B	05	06	00	00	00	00	c 'vú~QÓ.PK
00026130	02	00	02	00	BB	00	00	00	6D	60	02	00	00	00			»m`

📓 文件(F) 编辑(E) 搜索(S) 视图(V) 分析(A) 工具(T) 窗口(W) 帮助(H)

https://blog.csdn.net/gg_41668936

然后解压,发现flag.txt和Treeinblack.bmp,bmp格式经常是StegSolve有关的图片隐写格式,打开,然后一直点箭头,发现到 Red plane 3时候就已经出现结果。(说好的黑夜呢?不应该是black吗?)

File Analyse Help

Red plane 3



tips:

伪加密牵扯到zip压缩包格式问题,在zip中,504B0304为文件头(即文件标识),1400为解压所需版本,0100为加解密(偶不需要密码, 奇数需要密码,所以修改09为00),0800为压缩方式,后面的两个xxxx和xxx为文件修改时间和日期,再后面为CRC-32校验、尺寸、文件 名长度、扩展记录等等。

pikaqiu

Why can't I open this picture

解压后发现pikaqiu.jpg的图片,但是损坏,放到HxD里面,发现文件头和文件尾都很不正常,搜了下jpeg的文件头和文件尾,文件头为FFD8FF,文件尾为FFD9,添加即可,注意初始已经有FF了,所以文件头只需要添加FFD8。

tips:

不做搬运工,直接放链接文件头与文件尾总结

decode

see through the appearance to perceive the essence

解压,decode.docx的文件,估计是文档隐写,

在word中找到显示,把隐藏文字勾选即可。(复制时候可能无法复制,直接清除样式即可)

Word 选项		?	Y X
常规 显示	更改文档内容在屏幕上的显示方式和在打印时的显示方式。		
校对	页面显示选项		
保存 版式 语言	 ✓ 在页面视图中显示页面间空白(W)[□] ✓ 显示突出显示标记(H)[□] ✓ 最停时显示文档工具提示(L) 		
轻松访问 高级	始终在屏幕上显示这些格式标记		
自定义功能区 快速访问工具栏 加载项 信任中心	□ 制表符(1) → □ 空格(5) ・・・ □ 段潜标记(M) ・・・ □ 段潜标记(M) ・・・ □ 門选证字符(D) ・・・ □ 可选注字符(M) ¬ □ 可选介隔符(0) □ □ 可选分隔符(0) □ □ 显示所有格式标记(A) ·・		
	打印选项 ✓ 打印在 Word 中创建的图形(B) ^① 打印背景色和图像(B) 打印文档属性(P) 打印陶藏文字(X) 打印前更新城(F) 打印前更新链接数据(K)		v
		确定	

damage

Help me fix him 格式: Flag{}

还是图片隐写类,放到HxD里面修改文件头,文件头与文件尾总结,注意已经有G和9了,加上IF8(即49 46 38就行了),然后 是动图,简单,放到PS或者StegSolve——Frame Browser就行了(注意e可能会被认为是o)

低头才是王道

99 9 9 88 11 5 5 66 3 88 3 6 555 9 11 4 33 格式:jactf{}

提示低头,低头肯定是键盘密码了,连续的都是重复,而且不超过3个,所以第一个数字确定从哪数,也就是横坐标,多少个位数决定纵坐标。

所以99是l, 9是o, 88是k, 以此类推, 得到一串字符, 加上格式就行了

数据包分析

第一解出人: Gemini_Pulsar misc-挑战1

一堆数据表,还是走基本路子,大多在HTTP里面,Protocol排序后先看HTTP协议,发现一堆base64有关的,应该是base64编 码传输,复制一两个,然后解码

🚺 🔳 🖉 🥯 📙 🛅 🛾	🗶 🖾 🔍 🔄 👄 🛸	19 👲 📃 📃	କ୍ର୍ କ୍ 👖
---------------	-------------	----------	-----------

0170

0180

L.	应用显示过滤器 ⋯ 〈Ctrl-/〉					→	➡ 表达式… │ +
ο.	Time	Source	Destination	Protocol	Length Info		- ^
	1977 89.325980298	Vmware_c0:00:08	Broadcast	ARP	60 Who has	192.168.25.2? Tell	192.16
	2023 90.655724749	Vmware_c0:00:08	Broadcast	ARP	60 Who has	192.168.25.2? Tell	192.16
	2054 91.303486153	Vmware_c0:00:08	Broadcast	ARP	60 Who has	192.168.25.2? Tell	192.16
	2090 92.293732205	Vmware_c0:00:08	Broadcast	ARP	60 Who has	192.168.25.2? Tell	192.16
	2108 93.637230432	Vmware_c0:00:08	Broadcast	ARP	60 Who has	192.168.25.2? Tell	192.16
	2135 94.278780882	Vmware_c0:00:08	Broadcast	ARP	60 Who has	192.168.25.2? Tell	192.16
->-	180 8.260686231	192.168.25.128	192.168.43.83	HTTP	729 POST /u	pload/1.php HTTP/1.	1 (ap
-4	183 8.282645099	192.168.43.83	192.168.25.128	HTTP	399 HTTP/	展开子树(X)	Shift+Right
	187 8.486797464	192.168.25.128	192.168.43.83	HTTP	779 POST	展开全部(E)	Ctrl+Right
	189 8.491449977	192.168.43.83	192.168.25.128	HTTP	474 HTTP/	收起全部(A)	Ctrl+Left
	192 8.581128299	192.168.25.128	192.168.43.83	HTTP	779 POST		
	195 8.586889274	192.168.43.83	192.168.25.128	HTTP	474 HTTP/	应用为列	
	259 10.374211302	192.168.25.128	192.168.43.83	HTTP	771 POST	作为讨滤器应用	•
	264 10.391233224	192.168.43.83	192.168.25.128	HTTP	1182 HTTP/	准备过法器	
	267 10.393175840	192.168.25.128	192.168.43.83	HTTP	771 POST	「「日」と「時間	
	270 10.432822603	192.168.43.83	192.168.25.128	HTTP	1183 HTTP/	川山辺崎市	
	327 12.322104301	192.168.25.128	192.168.43.83	HTTP	769 POST	用过调谐者已	•
	331 12.340672586	192.168.43.83	192.168.25.128	HTTP	1311 HTTP/	追踪流	•
	33/ 10 3/5187100	102 168 25 128	102 168 /3 83	НТТР	760 DOST	复制	•
	[HTTP request 1/6]]				显示分组字节	
	[Response in frame	<u>e: 183]</u>				垦出公组字节流(B)	Ctrl+H
	[Next request in ·	<u>frame: 187]</u>				чтилят - гимо)	Currin
	File Data: 675 by	tes				Wiki 协议页面	
· +	ITML Form URL Encode	d: application/x-www	-form-urlencoded			过滤器字段参考	
	✓ Form item: "aa" =	"@evalD(base64_decod	<pre>le(\$_POST[action]));"</pre>			协议首选项	•
	Key: aa						
	Value: @eval\00	1(base64_decode(\$_P0	ST[action]));			解码为(A)	
	Form item: "action	n" = "QGluaV9zZXQoImF	<pre>Rpc3BsYX1fZXJyb3JzIiwi</pre>	MCIp00Bz	ZXRfdG1tZV9s	转至链接的分组	
	Key: action					在新窗口中显示已链接的	分组
	Value [truncate	d]: QGluaV9zZXQoImRp	c3BsYX1fZXJyb3JzIiwiM	CIpO0BzZ)	(RfdGltZV9saW1p	dCgwKItAc2V0X21hZ2	LJX3F1b3R1c1
			_				>
01	10 60 Ed 20 20 2h 1		2d 51 47 6c ml)).	loc tion-	001		-
01	75 61 56 39 7a	5a 58 51 6f /9 6d 50	$70 63 33 / 2$ $\mu / 97$		VGT VGB		
01	30 73 59 58 6c 66	5a 58 4a 79 62 <u>33 4</u> a	7a 49 69 77 sYX1f	$X_{\rm J} vb31_{\rm Z}$	Tiw		
014	10 69 4d 43 49 70	4f 30 42 7a 5a 58 52	66 64 47 6c iMCIp	DOB zZXRf	dG1		
01	50 74 5a 56 39 73 0	61 57 31 70 64 43 67	77 4b 54 74 tZV9s	aW1 pdCgw	ιKTt		
010	50 11 63 32 56 30	58 32 31 68 52 32 60	62 58 33 46 Ac21/01	(21 6721-	Y3E		

在线加密解密(采用Crypto-JS实现)

31 62 33 52 6c 63 31 39 79 64 57 35 30 61 57 31 6c 4b 44 41 70 4f 32 56 6a 61 47 38 6f 49 69 30

Feedback

月文:	BASE64:
@ini_set("display_errors", "0");@set_time_limit(0);@set_magic_quotes _runtime(0);echo("-> ");;\$D=dimame(\$_SERVER["SCRIPT_FILENAME [\$D]\t";if(sUbstr(\$D,0,1)="/"){foreach(range("A", "Z") as \$L)if(is_dir(" {\$L}:")\$R=""(\$L):";\$R="t";\$u= (function_exists(posix_getegid"))?@posix_getpwuid(@posix_geteuid()): ";\$usr=(\$u)?\$u["name]:@get_current_user();\$R.=php_uname();\$R.=" ({\$usr})";print \$R;;echo(" <");die();	QGluaV9zZXQolmRpc3BsYXIfZXJyb3JzliwiMClpO0BzZXRfdGltZV9saW 1pdCgwKTtAc2V0X21hZ2JjX3F1b3Rc19ydW50aW11KDApO2VjaG80i +fClpOzskRD1kaXJuYW1IKCRf00VSVkVSWyJTQ1JJUFRfkIMRU5E TUUiXSk7aWYoJEQ9PSliKSREPWRpcm5hbWUoJF9TRVJWRVJbllE BVEnfVFJBTINMQVRFRCJdKTskUj0ieyREfVx0ljtp2ihzdWJzdHloJEQ MCwxKSE9li8iKXtmb3JJYWNoKHJhbmdlKCJBliwiWilpIGFzICRMKWI KGIzX2RpcigieyRMfToiKSkkUi49InskTH06jt9JFluPSJcdCl7JHU9KGZ bmN0aW9uX2V4aXN0cygncG9zaXhf2V0ZWdpZCcpKT9AcG9zaXhf2 V0cHd1aWQoQHBvc2l4X2dIdGV1aWQoKSk6Jyc7JHVzcj0oJHUPyF 1WyduYW1JJ106QGdldF9jdXJyZW50X3VzZXloKTskUi49cGhwX3VuYN 1IKCK7JFluPSloeyR1c3J9KSI7cHJpbnQgJFl7O2VjaG8oInw8LSIpO2R ZSgpOw==

1b3Rlc19 ydW50aW1 1KDApO2V jaG8oIi0 百度了一下可能是中国菜刀一句话有关的东西,不过没关系,做这个题时候我也没学好菜刀,然后挨个复制了几个HTTP的 base64编码传输,大多都是这样,没啥用。开始针对性找,对base64的先忽略。

然后发现一个下图的数据包,后面有FFD8之类的,前面图片隐写知道这是图片的文件头,后面FFD9也对应,复制Form item:"z2"的值,放到HxD里面,并且重命名为jpg后缀,提交一下,发现不对,就知道没这么简单,那估计是密码。



继续向后走快结束HTTP时,发现下图的数据包,PK是zip压缩包的前缀,后面有well,you need passwd!,还有箭头,确定了,复制值(别复制编码后的,通过Line-based显示分组字节并改为原始数据再复制),然后解压输入上面图片得到的密码即可拿到有fag的flag.txt文件。

sqlmap二分法

类型: Forensic 黑客利用漏洞从Web系统中窃取了什么机密信息?

sqlmap二分法,幸亏之前学了sql注入,然后刷了sqli-labs,二分法就是通过盲注然后猜测数据库内信息的名称的过程。

web

crypto