【CTF】记录一次CTF比赛的Writeup(附题目下载地址)

原创

 TeamsSix
 ● 7 2019-09-25 17:16:15 发布
 ● 2122
 ◆ 收藏 12

 分类专栏:
 Writeup 文章标签:
 CTF 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_37683287/article/details/101374506

 版权

C

Writeup 专栏收录该内容

2篇文章0订阅 订阅专栏

0x00 前言

最近因为省赛快来了,因此为实验室的小伙伴准备了这次比赛,总共10道题目,考虑到大多数小伙伴都刚从大一升到大二,因此整体难度不高,当然有几道难度还是有的。

题目大多数都是从网上东找西找的,毕竟我也是个菜鸟呀,还要给他们出题,我太难了。

废话不多说,直接上Writeup吧,以下题目的文件下载地址可以在我的公众号(TeamsSix)回复CTF获取。

0x01 隐写 1

flag: steganoI

flag格式: passwd:

题目来源: http://www.wechall.net/challenge/training/stegano1/index.php

签到题,下载题目图片,利用记事本打开即可看到flag



0x02 隐写 2





打开图片,参考题目提示说一般在公共场合才能看见,因此通过盲文对照表可以得出flag是teamssix,图片中的AXHU只是用来 干扰的,这道题也是我参考wechall里面的一道题型。

а	b	~	d	e	f	g	h	i	j
• •	.0	••	••	• •	••	••	• •	0.	•
00	G	00	00	00	0 0	00	0 0	00	00
k	T I	m	n	o	р	q	r	s	t
30	• 0	••	••	• •	••	••	• 0	•	Q .
• 0	• 0	• 0	• •	• •	•••	• •	• •	• 0 • 0	• •
u	v	х	у	z					
• •	• 0	••	••	6					
••	• •	••							
		~	5						w
		2							•
	xe	0					0		••

0x03 Web 1

flag:iamflagsafsfskdf11223
Hint:站内有提示
题目地址:
http://lab1.xseclab.com/sqli2_3265b4852c13383560327d1c31550b60/index.php
参考来源: http://hackinglab.cn/ShowQues.php?type=sqlinject

1、打开题目地址

SQLi2	× +
← → ♂ ଢ	(i) 🎽 lab1.xseclab.com/sqli2_3265b4852c13383560327d1c31550b60/index.php
用户名: 密 码: 验证码:	te anssit.cu



2、查看源码找到提示

SQLi2 ×	http://lab1.xseclab.com/sqli2_32 \times +		×OO
← → ♂ ✿	i view-source:http://lab1.xsecla	b.com/sqli2_3265b4852c1338356	0327d1c31550b60/index.php
1 <html></html>	×CO	on.	
2 $\langle head \rangle$	~	G	
3 💉 <tit< td=""><td>:le>SQLi2</td><td>it.</td><td></td></tit<>	:le>SQLi2	it.	
{○		S	
.↓ Š <body></body>		di	^
5 (for	m action="" method="P	OST">	an'
7	用户名: <input <b="" type="te</td><td>ext"/> name="username"	e″> <br< td=""></br<>	
8	密 码: <inpu< td=""><td>t type="password"</td><td>name="password"></td></inpu<>	t type="password"	name="password">
9	验证码: <input <u="" type="te</td><td>ext″ name=″vcode″_></td><td>≪br≻ http:/		
10 <td>orm <!-- Tips login as</td--><td>admin></td><td>)-</td></td>	orm Tips login as</td <td>admin></td> <td>)-</td>	admin>)-
11 $\langle body \rangle$			1
12 >>		xO	-0
13	•	~	, C
CON .	n		GIT

3、根据提示使用admin登陆,并使用弱密码



4、尝试多次都提示失败,利用万能密码再做尝试,找到flag







0x04 Web 2

flag:76tyuh120KKytig#\$%^&
题目地址: http://lab1.xseclab.com/upload3_67275a14c1f2dbe0addedfd75e2da8c1/
flag格式: key is :
题目来源: http://hackinglab.cn/ShowQues.php?type=upload

1、打开题目地址,发现是一个文件上传界面



sit.com --147951099121753 Content-Disposition: form-data; name="file"; filename="TheOldCat.jpg" Content-Type: image/jpeg ams

3、在Burp中对文件名进行修改,比如在jpg后加上.png或者其他东西,成功看到flag

	~0 [°]		
Go Cancel < Y			
Request		Response	
Raw Params Headers Hex		Raw Headers Hex Render	
POST /upload3_67275a14c1f2dbe0addedfd75e2da8c1/upload_file.php HTTP/1.1 👔 🎽	A	HTTP/1.1 200 OK	
Host: lab1.xseclab.com		Server: nginx	\sim
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:69.0) Gecko/20100101 Firefox/69.0		Date: Fri, 20 Sep 2019 08:56:53 GMT	
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8		Content-Type: text/html; charset=utf-	
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		Connection: close	, U
Accept-Encoding: gzip, deflate		Via: 100142	₩.
Content-Type: multipart/form-data; boundary=147951099121753	C	Content-Length: 27	L ⁸
Content-Length: 127631		<u> </u>	
Connection: close		key is 76tyuh120KKytig#\$%^&	
Referer: http://lab1.xseclab.com/upload3_67275a14c1f2dbe0addedfd75e2da8c1/			
Cookie: PHPSESSID=54/180a6ea564c3d/1e4c41032d99f18		0.0	
Upgrade-Insecure-Requests: 1		×e0	C
1//7051000101752			
Contact Disposition: form data: name="file": filename="TheOldCatting name"			:+-
Content Disposition. Joint data, name- me , mename- me Oddat.jpg.phg			S
contentritipe, intageijkeg	G		S
	0.0.10011 16384Windows Photo		di
Editor 10 0 10011 163842019:03:23 22:35:540000000000000000000000000000000000	22:34:322019:03:23		N

0x05 soeasy

flag:HackingLabHdd1b7c2fb3ff3288bff Hint:在这个文件中找到key就可以通关 flag格式:key: 题目来源: http://hackinglab.cn/ShowQues.php?type=pentest

解法一:

1、下载文件后,发现是vmdk文件,利用DeskGenius打开后,发现Key,此为正确答案







解法二:

1、利用Vmware映射虚拟硬盘同样可以打开

映射或断开。	映射虚拟磁盘	×	N A
— 映射的虚拟 ₩	虚拟磁盘 选择要映射到i	+箕机驱动器上的虚拟磁盘文件以及相关联的卷。映射完成后,此驱动器即显	
96	示在 Windows	资源管理器内。	
	卷(⊻):	数 最大大小	
651		1 1019.7 MB	
	中中	□以只读模式打开文件(推荐)(图)	
	吸动到 驱动器(<u>D</u>):	Z: ····································	接(D)
	~	确定 取消 帮助	帮助



0x06 Crack

Hint:flag就是密码

题目: 邻居悄悄把密码改了, 你只知道邻居1994年出生的, 能找到她的密码吗?

题目来源: http://hackinglab.cn/ShowQues.php?type=decrypt

1、下载题目文件,根据题意,需要对WiFi密码破解,而且密码很有可能是邻居的生日,因此我们利用工具生成字典。

超级字典生成器			- • •
拼音字典 常规字典	电话号码 日期字典	姓名字典 英文单词	社会工程 弱口令
日期选择 日期范围: 1	994/1/1 🗐 🗸 -		
日期格式			
☑ 年月日(19990	526) 日年月日(1	999/05/26) 🗌 日月:	年(26051999)
□年月日(99052	6) 🗌 月日年(0	5261999) 🗌 日月:	年(260599)
年月日(9956)	□月日年(0	52699) 🗌 年月	(199905)
年月日(1999.	05.26) 🗌 月日年(0	05/26/1999) 🗌 年日	(199926)
□年月日(1999-	05-26) 🗌 月日年(0	05-26-1999) 🗌 月日	(0526)
□ 自定义日期	格式	例:	YYYY -MM -DD
□ 全选			
	生成字明	•	
状态: 设置起始日期	和终止日期,再选择日	期格式生成日期(生日)	字典

2、接下来利用ewsa进行破解,可以看到破解后的密码



这道题目当时实验室有人用kali做的,kali下的工具感觉破解速度更快。

0x07 BiliBili

题目:bilibili

flag格式: ctf{}

解法一:

1、使用Wireshark打开数据包,直接搜索ctf

	bilibili.pcap	:+	•		
文	‡(E) 编辑(E) 视图(⊻) 跳转(<u>G</u>) 捕获(<u>C</u>) 分析(A) 统计(S) 电话(Y) 无线(W)	工具(工) 帮助(н
	🔳 🖉 🕥 📙 🛅 🕻	🗙 🖻 । ९. 🗢 🗢 警 💽 .	🕹 📃 📃 ବ୍ ବ୍ 👳		
	应用显示过滤器 ··· ‹Ctu	r1-/>		•	
	分組详情 ~	寛宿 〜 □	区分大小写 字符串 > 。	f	
No.	Time	Source	Destination	Protocol I	Length Info
	1 0.000000	47.95.42.129	172.20.10.2	TLSv1.2	91 Application Data
	2 0.050486	172.20.10.2	47.95.42.129	TCP	54 60503 → 443 [ACK] Seq=1 Ack=38 Win=510 Len=0
	3 1.124936	172.20.10.2	106.75.240.103	тср	66 61136 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
	4 1.177651	106.75.240.103	172.20.10.2	TCP	66 80 → 61136 [SYN, ACK] Seq=0 Ack=1 Win=42420 Len=0 MSS=1360 SACK_PERM=1 WS=2048
	5 1.177736	172.20.10.2	106.75.240.103	ТСР	54 61136 → 80 [ACK] Seq=1 Ack=1 Win=131840 Len=0
	6 1.178219	172.20.10.2	106.75.240.103	HTTP	716 GET /148389186 HTTP/1.1
	7 1.225638	106.75.240.103	172.20.10.2	TCP	54 80 - 61136 [ACK] Seq=1 Ack=663 Win=45056 Len=0
+	8 1.252101	106.75.240.103	172.20.10.2	тер	1414 80 - 61136 [ACK] Seq-1 Ack-663 Win-45056 Len-1360 [TCP segment of a reassembled PDU]
	9 1.257885	106.75.240.103	172.20.10.2	HTTP	1007 HTTP/1.1 200 OK (text/html)
	10 1 257993	172 20 10 2	106.75.240.103	TCP	54 61136 - 80 [ACK] Soq-663 Ack-2314 Win-131840 Lon-0
	11 1.277528	172.20.10.2	120.92.150.127	TLSv1.2	91 Application Data
	12 1.318071	120.92.150.127	172.20.10.2	TLSv1.2	85 Encrypted Alert
	13 1.318071	120.92.150.127	172.20.10.2	TCP	54 7823 → 60426 [FIN, ACK] Seq=32 Ack=38 Win=63 Len=0
	14 1.318160	172.20.10.2	120.92.150.127	TCP	54 60426 → 7823 [ACK] Seq=38 Ack=33 Win=510 Len=0
-	5 1.334500	172.20.10.2	175.6.237.122 👅 🧷	TCP	66 61148 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
~	16 1.378900	172.20.10.2	120.92.150.127	TCP	54 60426 → 7823 [FIN, ACK] Seq=38 Ack=33 Win=510 Len=0
	17 1.383969	175.6.237.122	172.20.10.2	TCP	66 80 → 61148 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1360 SACK_PERM=1 WS=512
	18 1.384118	172.20.10.2	175.6.237.122	TCP	54 61148 → 80 [ACK] Seq=1 Ack=1 Win=131840 Len=0
	19 1.384577	172.20.10.2	175.6.237.122	HTTP	517 GET /bfs/seed/jinkela/header.js HTTP/1.1
	20 1.433888	120.92.150.127	172.20.10.2	тср	54 7823 → 60426 [RST] Seq=33 Win=0 Len=0
	21 1.433889	175.6.237.122	172.20.10.2	TCP	54 80 → 61148 [ACK] Seq=1 Ack=464 Win=30720 Len=0
	22 1.434299	175.6.237.122	172.20.10.2	HTTP	1149 HTTP/1.1 304 Not Modified
	23 1 475546	172 20 10 2	175 6 237 122	TCP	54 61148 - 80 [ACK] Sog-464 Ack-1006 Win-130816 Lon-0

2、找到标识的那一行右击进行追踪对应的协议,比如这条是http协议就追踪http协议,之后再次查找ctf

▲ Wireshark · 追踪 HTTP 流 (tcp.stream eq 1) · bilibili.pcap	-		×
if (ua indexOf(agents[i]) > -1) {			^
window.location.href = 'https://m.bilibili.com/space/' + mid			
break			
} <link as="script" href="//s1.hdslb.com/bfs/static/player/main/video.js?v=2019827" rel="prefetch"/> <script type="text/javascript</td"><td>snc=//</td><td></td><td></td></tr><tr><td>s1.hdslb.com/bfs/static/jinkela/long/js/sentry/sentry-5.2.1.min.js></script> <script src="//s1.hdslb.com/bfs/static/</td" type="text/javascript"><td>jinkela,</td><td>/long/</td><td></td></tr><tr><td>js/sentry/sentry.vue.js></script> <link href="//s1.hdslb.com/bfs/static/jinkela/space/css/space.4.cd1154764e06b09a30dad765d2f68dfc57c172c</td"/> <td>2.css</td> <td>0.</td> <td></td>	2.css	0.	
rel=stylesheet> <link href="//s1.hds1b.com/bfs/static/jinkela/space/css/space.3.cd1154764e06b09a30dad765d2f68dfc57c172c2.css" rel="styleshe</td"/> <td>et></td> <td></td> <td></td>	et>		
<title>TeamsSix () A Bilibili</title> <meta <="" name="keywords" td=""/> <td></td> <td></td> <td></td>			
content="TeamsSix,B,AMV,MAD,MTV,ANIME,,AMV,MAD,MTV,ANIME,,ACG,galgame,,,ACG,galgame,,ACG,galgame,,,ACG,galgam		,voca	
loid"/>			
<meta <="" name="description" td=""/> <td></td> <td></td> <td></td>			
content="TeamsSixctf{Q29uZ3JhdHVsYXRpb25zX31vdV9nb3RfaXQ=}bilibili			
ACG			
<pre><body><div_class="z-top-container_has-top-search"><body><div_id="browser-version-tip"></div_id="browser-version-tip"></body></div_class="z-top-container_has-top-search"></body></pre>			
<pre></pre>			
<pre></pre>	html">.		
Chrome			
		2	
endif? <div id="space-app"></div> <script type="text/javascript"></script>			



解法二:

1、和解法一一样,对数据包进行追踪http流,不难看出这是访问space.bilibili.com/148389186的一个数据包



2、打开这个网址,同样可以看到被base64加密的flag





另外打个小广告,上面这个是我的bilibili号(TeamsSix),欢迎大家关注,嘿嘿

0x08 Check

flag:sAdf_fDfkl_Fdf
题目:简单的逆向
flag格式: flag{}
题目来源: https://www.cnblogs.com/QKSword/p/9095242.html

1、下载文件,发现是exe文件,放到PEiD里看看有没有壳以及是什么语言编写的,如果有壳需要先脱壳。

🚟 PEID v0.95			-		×			
文件: D:\Users\theoldcat\Desktop\check.exe								
入口点: 0000135%		EP 段:	. text		\triangleright			
文件偏移: 0000075E		首字节:	E8, DE,	, 02, 00	\sum			
连接器版本: 11.0	T	子系统:	Win32	console	\triangleright			
PESniffer:					$\mathbf{\Sigma}$			
PEiDDSCAN: Microsoft Vi	sual C++ 8 *				\sum			
多文件扫描(11) 2	至看进程(工)	扩展信息		插件				
✓ 总在最前(S)	选项(0)	关于(A)		退出				

2、可以看到使用的C语言写的,同时是32位,因此使用IDA32位打开,之后找到main函数

🦅 IDA - check.exe C:\Users\theoldcat\Deskto	p\check.exe					
文件(F) 编辑(E) 跳转(J) 搜索(H) 视图(V)	文件(F) 編編(E)					
2 🛋 🗖 2 👍 🗸 🔿 🖉 8 🏔 🗛 🛝 🛝 🗋	6 1	A :	* -* -* -* -* -* -* -* -	于调计器	- 🐟 🗈	🖅 🐏 🔍
	3×0 ; 🛋	- 1: 19		20090468		
■ 库函数 ■ 常规函数 ■ 指令 ■ 数据 ■ 未	知 外部符	묵				
☑ 函数窗口		ēΧ	TDA View-A		估任四	
				453	NAL NHA	
遊戲名称	段	ŧ ^				. HOUEL ITAC
🗾 _main	. text	0				
	. teat	0				; Segment type: Pure code
<pre>security_check_cookie(x)</pre>	.text	0				; Segment permissions: Read/Execute
f _pre_c_init	text	0				.text segment para public 'CODE' use32
f _pre_opp_init	. text	0				assume cs:.text
Jtmainchistartup	text	0				source estimating scienting der data fernothing genothing
f raise securityfailure	towt	0				assume estimetizing, sstinetizing, astroacti, istinetizing, gstinetizing
f report gefailure	text	0				
CxxUnhandledExcentionFilter(EXCEPTI	text	0				; Attributes: bp-based frame
7 sub 4014DF	text	0				
7_XcptFilter	. text	0				; intcdecl main(int argc, const char **argv, const char **envp)
📕 _amsg_exit	. text	0				_main proc near
f sub_4014FA	. text	0				Buf- bute ptc -108b
FindPESection	. text	0				var 4= dword ptr -4
IsNonwritableInCurrentImage	.text	0				argc= dword ptr 8
📝ValidateImageBase	. text	0				argv= dword ptr 0Ch
fsecurity_init_cookie	. text	0				envp= dword ptr 10h
<u>f</u> atonexitinit	. text	0				
<u>f</u> onexit	.text	0				push ebp
f_atexit	. text	0				mov ebp, esp
f sub_401701	text	0				sub esp, ivon
7 sub_401/P1	. text	0				xor eax, ebp
jsetderaditprecision	text	0				mov [ebp+var 4], eax
je	text	0				push offset Format ; "输入的flag为"
F SEH prolog4	text	0				call ds:printf
F SEH epilor4	text	0				lea eax, [ebp+Buf]
F except handler4	. text	0				push 104h ; Size
F ort debugger hook	text	0 ¥				push eax ; our
<		>				



add	esp, øun
lea	ecx, [ebp+Buf]
call	sub_401050
mov	ecx, [ebp+var_4]
xor	ecx, ebp
xor	eax, eax
call	<pre>@security_check_cookie@4 ;security_check_cookie(x)</pre>
mov	esp, ebp
non	ebp

3、按F5查看伪代码,并点击sub_401050子函数



4、不难看出下列是一个10进制到ASCII码的转换



5、利用在线网站转换即可获得flag,网站地址:http://ctf.ssleye.com/jinzhi.html

文本	flag{sAdf_fDfkl_Fdf}	nssit.	teams	
二进制	01100110 01101100 01100001 0110	清 9 0111 01111011 01110011 01000001 0110010	空 100 01100110 01011111 01100110 01000100	0
らけ、 +进制	102 108 97 103 123 115 65 100 102	95 102 68 102 107 108 95 70 100 102 125		



0x09 Android RE

flag:DDCTF-397a90a3267641658bbc975326700f4b@didichuxing.com
题目:安卓逆向
flag格式: DDCTF-
Hint:flag中包含chuxing
题目来源: https://xz.aliyun.com/t/1103

1、这道题是滴滴出行的一道CTF,下载题目可以看到一个apk文件,先在模拟器中运行看看是个什么东西



2、功能很简单,一个输入框,输错会提示Wrong,那么利用Android killer给它反编译一下,查找字符"Wrong"

		- 25		Android Kill	er V1.3.1.0	– 🗆 X
	〔 主页 〕	IĮ	Android	XO	~O``	0
ジ 打开 文作	↓ ② F 隐藏面 板 机图	武 武 武 近 项	 关于 帮助 	com	assit.c	
a 3	开始 🎽 DD	CTF-Norm	nal	it.	all.	~
				MainActivity.smali	×O	
	工程信息	工程管理	器 🖸 工程搜索	፩ ≡ ◙ (≌ (⇔) 🖓 🗈 🗈 3		°
<u>_</u> #	體索字符: ——		-0	MainActivity.smali		· ·
	wrong		xOV	67 const-string v1 "Correct"	est.	^

建集 諸決字符: 重拾字符: 全部替换 建築范碼: 建築范碼: 当前整个项目 文件类型: Ismali(xml),btd,htm,l.html 字符编码:	<pre>invoke-virtual {v0, v1}, Landroid/widget/TextView;->setText(Ljava/lang/CharSequence;)V .line 53 .goto_0 return.void ine 51 ;cond_0 iget-object v0, p0, Lcom/didictf/hellolibs/MainActivity;->mFlagResultView;Landroid/widget/TextView; const-string v1, "dicods" invoke-virtual {v0, v1}, Landroid/widget/TextView;->setText(Ljava/lang/CharSequence;)V goto :goto_0 .end method .method protected opCreate(Landroid/os/Bundle;)V .locals 1 .param p1, "savedInstanceState"</pre>	5
UTF8	行: 79 列: 28 插入	20
■ 匹配大小写		0
搜索进度: teamsst.	♥ wrong ♥ smali\android\support\v4\content\Loader.smali ● smali\com\didictf\hellolibs\MainActivity.smali ● const-string v1, *Wrong*	

- 3、可以看到Wrong字符的路径,接下来进行反编译,不过可能由于本身软件的文件,反编译提示未找到对应的APK源码,没关
- 系,换ApkIDE对其进行编译

it.	Android Killer V1.3.1.0	
主页 工具 Android	xÕ	0
(回) ((U) (U) (U) </th <th>com ssit.</th> <th></th>	com ssit.	
合 开始	4.	~
	MainActivity.smali	
IT程信息 工程管理器 工程搜索 搜索字符: ····································	<pre>MainetonyJatas MainetonyJatas MainetonyJatas</pre>	ava/lang/CharSequence;)V ;ResultView:Landroid/widget/TextView; ava/lang/CharSequence;)V
搜索进度: SSIT-	<pre>wrong wrong const-string v1, "Attempting to unregister the wrong listener" const-string v1, "Attempting to unregister the wrong listener"</pre>	com
teall	smali\com\didictf\trellolibs\MainActivity.smali	sit
	日志輸出「提案結果」方法引用	offi





4、等待一段时间后,可以看到对应源码,简单分析就可以知道该代码从hello-libs.so文件加载,并且对 mFlagEntryView.getText().toString()函数的内容即我们输入的内容和stringFromJNI()函数的内容做判断,如果一致就Correct,即 正确,不一致就返回Wrong,即错误,那么接下来只需要分析stringFromJNI()的内容就行了,因此我们需要知道系统从hellolibs.so文件加载了什么



5、将APK解压,找到hello-libs.so文件,由于现在手机都是用arm64位的CPU(我也不知道是不是的啊,听别人说的),因此我 们找到larm64-v8a文件夹下的libhello-libs so文件, 田IDA打开

DDCTF-Normal → lib →	arm64-v8a	xÕ		
<u>^</u>	名称	修改日期	类型	大小
*	ibgmath.so	2019/9/21 11:40	SO 文件	6 KB
	libgperf.so	2019/9/21 11:40	SO 文件	6 KB
	libhello-libs.so	2019/9/21 11:40	SO 文件	6 КВ

其中, mips、armeabi、armeabi-v7a和x86都表示CPU的类型。一般的手机或平板都是用arm的cpu, armeabi 是针对普通的或旧的arm v5 cpu, 32位 armeabi-v7a 是针对有浮点运算或高级扩展功能的arm v7 cpu, 32位 arm64-v8a 针对64位的 mips 是一种采取精简指令集 (RISC) 的处理器架构, 32位 mips64 64位 x86 IA-32位指令集

6、打开IDA后,根据题目提示,Alt+T 查找chuxing

	👔 IDA - libhello-libs.so C:\Users\theoldcat\Desktop\DDCTF-Normal\lib\arm64-v8a\libhello-libs.so														
文	件(F) 编辑(E) 跳转(J) 搜索(H) 视图(V)	调试器	选项(O)	窗口(W) 帮助								in the second se	-		
i r	🗲 🔚 🗢 🕶 🚽 🍓 🛍 🍓 🖡		A 🥥 🗄	at at at 🖈 🖈	úХ			无调试器			•	1 12 12 12 12	×		
1								+							
1				1.								<u> </u>			<u> </u>
1	库函数 📕 常规函数 📕 指令 🔲 数据 📕 未知	印 🦳 外音	部符号	ca'								~0.			5
f	函数窗口		⊐ ₽ ×(IDA Viev	r-A		×	\bigcirc		十六进制	视图−1			结构体
ø	数名称	段	記始	0A600000000000000	E8 03	43 39	E9 03	03 91	0A 00	80 12	08 7D 01 53	····.9}.9	5		5
Ŧ	GetTicks	 n]+	0000	0000000000000980	2B 01	08 8B	E9 7B	1F 32	6C 15	40 38	29 05 00 91	+21.@8)		. 4	•
J F	ove finelize		0000	000000000000000000000000000000000000000	4A 05	00 11	AC FF	FF 35	51 05	00 /1	48 01 00 54	J5qK			-
f	stack chk fail	nlt	0000	000000000000000000000000000000000000000	29 /D E0 23	40 92	EA 03	14 00	34 05 35 55	66 91 FF 97	41 01 08 88 88 75 40 03)}@4A	•	20	
f	spower	nlt	0000	000000000000000000000000000000000000000	02 00	00 14	E8 03	1F ΔΔ	E9 23	00 91	3F 69 28 38	21(8	2	25	
f	. android log print	plt	0000	00000000000000000000000000000000000000	68 02	40 69	E1 23	00 91	EØ Ø3	13 AA	08 9D 42 F9	h.@B.			
f	.memcpy	.plt	0000	00000000000000A10	00 01	3F D6	48 DØ	3B D5	08 15	40 F9	E9 03 40 F9		0		
f	. cxa atexit	.plt	0000	00000000000000A20	1F 01	09 EB	C1 00	00 54	F3 7B	5A A9	F5 53 59 A9	·		P	
f	start	. text	0000	00000000000000A30	FC 5B	58 A9	FF C3	06 91	CØ Ø3	5F D6	1D FF FF 97	.[x	XO		
f	sub_70C	. text	0000	00000000000000A40	39 63	6A A6	00 31	55 3D	A3 60	EØ 91	6A 66 B3 6B	9cj1U=.`f.			
f	aeabi_wind_cpp_pr45(char *)	. text	0000	00000000000000A50	A9 0B	72 42	28 87	7C 8F	C1 E2	7E 4F	44 44 43 54	rB(. ~ODDC1	Г		
f	Java_com_didictf_hellolibs_MainActivit…	.text	0000	000000000000000A60	46 2D	33 39	37 61	39 30	61 33	32 36	37 36 34 31	F-397a90a3267641	1		
f	GetTicks	extern	0000	000000000000000A70	36 35	38 62	62 63	39 37	35 33	32 36	37 30 30 66	658bbc9753267001			
f	cxa_finalize	extern	0000	000000000000000000000000000000000000000	34 62	40 64	69 64	69 63	68 /5	78 69	6E 67 2E 63	4b@alalchuxing.c			
f	stack_chk_fail	extern	0000	000000000000000000000000000000000000000	9C 48	74 20	02 20 0C 20	95 12	67 54	GA 10	29 FD 4A 03 59 70 D7 50	V7= - gTi			5
f	gpower	extern	0000	000000000000000000000000000000000000000	57 7A	R3 6D	4D BC	60 1Z	07 54 A0 60	DA 19	10 15 BE 05	.KZ=gij			
f	android_log_print 🛛 🔷 🦳	extern	0000	000000000000000000000000000000000000000	02 64	84 72	7D 95	AD 6C	SE ED	42 (6	6E 5B 10 DD	dr} 1^ B [\sim
f	певсру	extern	0000	00000000000000000000000000000000000000	B3 F7	73 AD	1E 65	35 ØA	52 37	25 32	60 8B 26 FD	e5`.&			. 0.
f	cxa_atexit	extern	0000	00000000000000AE0	D8 C2	6B 42	82 67	C8 4D	7A 95	E8 81	48 C1 9E 40		- D		XO
	.0			00000000000000AF0	E8 FB	CF E6	4F BA	E6 AF	78 19	6F 9C	E9 F7 7A DD				
	• • •			000000000000000000000000000000000000000	42 CE	8C 03	B8 66	D3 AB	00 7E	DE 3E	53 DE 30 91	. BOfö~S			
				0000000000000B10	3D F7	CD 72	14 51	82 EE	18 8D	B4 8C	DØ 8A F6 9A	. =Qњ			
				0000000000000B20	96 71	98 62	93 4A	30 2F	9C A8	79 16	C1 E0 EC D7	'.q.b.J0/y			
	~~~~			0000000000000B30	E5 EC	8A 64	B4 46	CF D9	E5 96	F3 94	73 A9 FF EA	d.Fs			
				0000000000000B40	CB 15	9C 7C	A1 D8	3E BB	1D 38	CB 55	DØ 19 25 B2				
				0000000000000850	ØB 92	E8 88	AE 06	A2 9B	93 64	5E FB	09 05 F6 2F	舠··d^			
	20			000000000000B60	1F 35	CC EF	05 6C	19 42	38 A5	59 2E	80 0A 19 FC	.51.B8.Y			
				INGRAGAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	33 5R	BB D6	FR 2R	ΔC F7	ØF AD	NR 57	40 98 71 <b>2</b> 6	'≣31⊾" + Wa∩			

7、成功找到flag(DDCTF-397a90a3267641658bbc975326700f4b@didichuxing.com

#### )输入到模拟器中看到提示Correct,说明flag正确。



### 0x10 Easy_dump

flag: F0rens1cs_St2rt
题目: Easy_dump
flag格式: LCTF{}
Hint: volatilty了解一下
题目来源: https://www.tr0y.wang/2016/12/16/MiniLCTF/index.html

#### 解法一:

1、下载题目文件,提示利用volatilty工具,同时结合文件后缀为vmem(VMWare的虚拟内存文件),因此判断是一个内存取证的题目,关于volatility的使用可以参考官方手册:https://github.com/volatilityfoundation/volatility/wiki/Command-Reference,废话不 多说,先看看镜像信息

# volatility -f xp.vmem imageinfo

root	kali:~/Deskto	p# volatility ·	f xp.vmem ima	geinfo	9		2
Vola	tility Foundat	ion Volatility	Framework 2.6	, 			
INFO	: volatili	ty.debug : [	Determining pr	ofile based	on KDBG sea	rch	
	Suggeste	d Profile(s) :	WinXPSP2x86,	WinXPSP3x86	(Instantiate	ed with	WinXPSP2x86)
0		AS Layer1 :	1A32PagedMemo	гугае (кегле	LAS)		
		AS Layer2 :	FileAddressSp	ace (/root/[	esktop/xp.vr	nem) (+	
		PAE type :	PAE				
		DTB :	0xb2a000L				
		KDBG :	0x8054e2e0L				
	Number o	f Processors :	2				
	Image Type (S	ervice Pack) 🙁	3				
	KP	CR for CPU 0 :	0xffdff000L				
	KP	CR for CPU 1 :	0xfc72b000L				
	KUSER	SHARED DATA :	0xffdf0000L				
	Image d	ate and time :	2016-12-02 11	:17:55 UTC+0	0000		
	Image local d	ate and time :	2016-12-02 19	:17:55 +0800	)		XX

2、可以看到该镜像信息的为WinXPSP2x86,接下来直接扫描查看一些系统文件中有没有flag文件

# volatility -f xp.vmem --profile=WinXPSP2x86 filescan | grep flag

root@kali:~/Desktop#	volatility	-f xp.vr	memprofile=WinXPSP2x86 filescan   grep flag
Volatility Foundation	Volatilit	v Framewo	ork 2.6
0x00000000005ab74c8		9 RW-r	\Device\HarddiskVolumel\Documents and Settings\Administrator\桌面\flag.txt
0x0000000007782ef8	1	9 RW-rw-	\Device\HarddiskVolume1\Documents and Settings\Administrator\Recent\flag.lnk

3、将该flag.txt文件dump下来

# volatility -f xp.vmem --profile=WinXPSP2x86 dumpfiles -Q 0x000000000005ab74c8 -D ./ -u

root@kali:~/Desktop# volatility -f xp.vmem --profile=WinXPSP2x86 dumpfiles -Q 0x0000000005ab74c8 -D ./ -u Volatility Foundation Volatility Framework 2.6 DataSectionObject 0x05ab74c8 None \Device\HarddiskVolume1\Documents and Settings\Administrator\桌面\flag.txt

#### 4、直接cat flag文件即可看到flag



解法二:

因为该题作者将flag复制到了自己电脑的粘贴板里的,所以直接获取粘贴板的内容也是可以看到flag的,不过谁能想到这种操作 [笑哭]

# volatility -f xp.vmem clipboard

<mark>root@kali</mark> :∼ Volatility	/Desktop# vola Foundation Vol	atility -f xp.vmem cli latility Framework 2.0	ipboard 5			
Session	WindowStation	Format	Handle	0bject	Data	
0	WinSta0	0xc009L	0x7e009d	0xe1577c30		
O	WinSta0	CF_UNICODETEXT	0x1c00d1	0xe10ca7c0	LCTF{F0rens1cs_St2r	t}
Θ	WinSta0	0xc013L	0x1000b3	0xe1260b38		_
Θ	WinSta0	CF LOCALE	0x16008b	0xe15a08e8		
Θ	WinSta0	CF_TEXT	0x1			
Θ	WinSta0	CF_0EMTEXT	0x1			

以上就是本次我为他们准备的CTF的全部内容,大多数都是很基础的题目,平时拿来练练手还是不错的,拓宽一下自己的了解 面,发现一些自己以前不知道的东西,如果你也想拿上面的题目来玩玩,在公众号(TeamsSix)回复CTF就可以获取下载地址 哦。 更多信息欢迎关注微信公众号: TeamsSix 原文链接: https://www.teamssix/year/190925-114420.html