各类移动安全竞赛题/部分writeup收集与整理



9 篇文章 1 订阅 订阅专栏

各类移动安全竞赛题/部分writeup收集与整理

小编偶然发现在google云端网盘上Mobile Challenges有各个网友上传的关于移动安全的题 目https://drive.google.com/folderview?id=0B7rtSe_PH_fTWDQ0RC1DeWVoVUE&usp=sharing,有需要的可下 载。

我的是	云端硬盘 > Mobile	2	G	**	•		ĄĴZ	0	\$			
	oco호스밴ᅣ호	 Alibete 001512		ALCOTE ODD A			A sis OTT	0014			075 001	
	360女主挑成恭 …	Allbaba 2015to		AllCTF 2014			ASIS CTF	2014	 8	Atast	CTF 201	Z
	BCTF 2014	COCON CTF 20		CircleCityCon 2···			CodeGat	e CTF ···		Crack	mes.de	
	CSAW CTF 201····	CSAW CTF 201····		CUIT 2014			Cyber Se	curity ···		DEFK	THON C	т
	DFIRCON EAST ···	http:/ EnoWars CTF 2…	/b1o	g. csdn. net/pi GreHack CTF 2…		be_2	2012 Hack.Lu	CTF 20		Hack	Lu CTF :	20
	HackPlayer	HCTF 2014		Insomni'hack 2····		Þ	JCTF 20	14 金…		MWR	BSides 2	20
	NDH2k14 warg····	No cON Name ····		Nuit du Hack 2k…			Nullcon	HackIM•••		Pico	CTF 2014	ł
	Pragyan CTF 20···	Qiwi CTF 2014		Reddit CTF 201····			Seculnsi	de CTF…		SU-C	TF 2014	

还收集了部分竞赛中一些牛人写的解题思路,下载网址如下:http://pan.baidu.com/s/1ntrDgc5 密码: byis

名称	修改日期
📕 360竞赛题.zip	2015/6/2 20:57
📑 2014AliCTF前三名解题思路.zip	2015/6/2 20:42
📑 2014攻防对抗挑战赛——NAGA PT& PIOWIND.zip	2015/6/2 20:47
📑 2015阿里巴巴安全竞赛.zp. csdn. net/phoel	2015/6/2-20:50
📑 2015阿里巴巴移动安全挑战赛.zip	2015/6/2 20:54
📑 Mobile_Challenges-2015-03-23.zip	2015/3/24 13:37
📄 其他竞赛通关攻略.txt	2015/5/5 22:28

以下是小编参加2015阿里组织的移动安全挑战赛时写的第二题解题思路,那时刚学android逆向,比较菜,各位大神请绕过~

方法一:

运行程序后没有log信息,又看了下源码,它加载了crackme.so动态链接库,调用了其中的函数 securityCheck(str),根据该函数的返回值(0或1)来判断是否输入正确。

首先进行静态分析crackme.so,载入IDA,查看securityCheck函数的源码:

```
signed int __fastcall Java_com yaotong_crackme_MainActivity_securityCheck(int a1, int a2, int a3)
  int v3; // r5@1
 int v4; // r4@1
 int v5; // r005
 char *v6; // r205
 int v7; // r3@6
 signed int v8; // r107
 v3 = a1;
 v4 = a3;
 if ( !byte_6359 )
  -
    sub_2494((int)&unk_6304, 8, (int)&unk_446B, (int)&unk_4468, 2u, 7);
    byte_6359 = 1;
  - 3
  if ( !unk_635A )
  {
    sub_24F4(&unk_636C, 25, &unk_4530, &unk_4474);
                                                   n.net/phoebe 2012
    unk_{635A} = 1;
  }
  _android_log_print(4, &unk_6304, &unk_636C);
 u5 = (*(int (__fastcall **)(int, int, _DWORD))(*(_DWORD *)u3 + 676))(u3, u4, 0);
u6 = off_628C;
  while (1)
  Ł
    v7 = (unsigned
                     int8)*vó;
    if ( v7 != *(_BYTE *)v5 )
     break;
    ++v6;
    ++v5;
    v8 = 1;
    if ( 107 )
      return v8;
 з
 return 0;
B
```

得知字符串v6所存的就是密码。查看偏移628c处所存的字符串,是"wojiushidaan"。输入看看,错误。好吧,我也觉得没有那么简单。估计是程序运行后把字符串改了。

.data:0000628C off_628C	DCD aWojiushidaan	; DATA XREF: Java_com_yaotong_crackme_MainActivity
.data:0000628C	1.1.1.7/1.1	; .text:off_130810
.data:0000628C ; .data	endSttp://blog.cs	dn. petwoplushDdaan012

刚学了android动态调试(http://www.52pojie.cn/thread-293648-1-1.html),就试试看,在JNI_OnLoad函数上下断点。

(1) 执行android_server: adb shell /data/local/tmp/android_server

- (2) 端口转发adb forward tcp:23946 tcp:23946
- (3) 调试模式启动程序adb shell am start -D -n com.yaotong.crackme/.MainActivity
- (4) IDA附加
- (5) 设置调试选项
- (6) F9运行
- (7) 执行jdb -connectcom.sun.jdi.SocketAttach:hostname=127.0.0.1,port=8700
- (8) 静态找到目标函数对应所在模块的偏移地址

JNI_OnLoad函数: 0x00001B9C

Java_com_yaotong_crackme_MainActivity_securityCheck: 0x000011A8

aWojiushidaan: 0x00004450

- (9) Ctrl+S找到libcrackme.so基地址0x4924C000,分别与上述三个地址相加得到最终地址。
- (10) G跳转至两个函数地址,然后下断点
- (11) F9运行
- (12) 断下,进行单步调试F8,同时查看0x49250450处的字符串。

 49250450
 77
 6F
 6A
 69
 75
 73
 68
 69
 64
 61
 6E
 00
 4C
 37
 39
 wojiushidaan.L79

 49250460
 00
 F4
 51
 0D
 75
 AC
 53
 00
 33
 66
 09
 41
 3C
 5B
 0B
 5B
 ...Q.u.S.3f.A<[.[</td>

 49250470
 81
 53
 9F
 00
 47
 56
 4D
 00
 73
 21
 23
 4C
 00
 42
 4D
 54
 .S..GVM.st#L.BMT

 49250480
 80
 A9
 B5
 B8
 B7
 C1
 55
 80
 60
 37
 90
 91
 49
 34
 72
 63
 ...U.m7..14r

 49250490
 4E
 B8
 80
 50
 53
 80
 39
 58
 2C
 2B
 28
 58
 3D
 21
 29
 CE
 N..PS.9X,+(X=!)
 ...

当单步调试JNI_OnLoad函数到某一步时,发现字符串变为"aiyou,bucuoo"。

	•	lib	cra	ckm	e.s	o:4'	924	DC54	SU	B				R	0,	R11	, t	t-va	ar	20		
	•	lib	cra	ckm	e.s	o:4'	924	DC58	BL	х				R	7				_			
	•	lib	cra	ckm	e.s	o:4'	924	DC5C	BL					u	nk	492	4D7	/F4				
	PC •	lib	cra	ckm	e.s	o : 4'	924	DC60	LD	R				B	0.	FR4	1					
	•	lib	cra	ckm	e.s	o:4'	924	DC64	MO	U				R	6.	#4	1					
		lib	cra	ckm	P.51	n : 4'	924	DC68	МО	Ŭ				R	1.	R5						
		lib	cra	c kim	0 4 5 4	n e M	024	0000	AR	Rim.					6	R6 1	et c	3v 1)		0		
		110	CF G	ekm	սպ	7.7	147	0070	MO	un.				Den.	<u>, </u>	D6	- 27		000			
		110	cra	ekm		n - hi	724	0070	1.0	Ď				D.	,	r d a	#1	10-14	1			
		110	LT d	CKIII	e.s.		924 096	V674 6070	MO	n 11				 	а, а	רעה היי	,#•	97.11	0 J			
		110	ura	CKIII	2.5	0:4	924	DC/0 D070	110					n n	υ,	K4						
		110	cra	скт	e.s	D:4	924	0676	BL	A 				- K	3							
		110	cra	CKM	e.s	D:4	924	DC80	UM	P				К	U,	#0						
6025 QJ:1	0 Ju	2 00	0.0	E5	89	8.0	RD	FQ	89	Ъß	20	FO	80	FF	FF	ER	н	1		-	a_	
4723041 6095069		000	0.0	55	87	20	60	EE	82	81	00	ER	80	00	0.0	E9	- 7					
4727042 6005060	0 4	6 30 6 66	20	E2	00	20	00	E2 E0	92	50	63	EU	00	00	00	E2			· · .	••••		
4925043	0 0	8 80 D E 0	BD	Eð	08	40	20	E9	30	FJ	FF	EB	08	40	20	EY	1		@			.@
4925044	0 3	8 F3	FF TO	EB	00	00	00	00	00	00	00	00	00	00	00	00	- 7	277	· • •			
4925045	<mark>0</mark> 6	69	79	61	75	20	62	15	63	15	6F	0F	មម	40	37	39	ം	190)u,I	buc	uoo	.L/
4925046	00	0 F4	51	ØD	75	AC	/53	-005-	33	66	00	41	30	'5B'	OB	-5B		.Q.	.u.:	S.3	F.A	۲.J>
4925047	00	1 53	9F	00	47	56	4D	00	73	21	23	40	00	42	4D	54		s	. GVI	M.s	!#L	. BM1
4925048	0 0	0 A9	B5	B8	B7	C1	55	00	6D	37	00	01	49	34	72	03			I	U.m	7	[4r.
4925049	0 4	E 88	00	50	53	00	39	58	20	2B	28	58	3D	21	29	CE	N	I P	'S . !	9X,	+ (X:	=1).

结果: aiyou, bucuoo

方法二:

dump内存。

程序运行在手机上(经测试,模拟器上貌似不行)。

通过dd命令,将内存里的数据拷贝出来。

dd if=/proc/6821/mem of=/sdcard/mem bs=1skip= 1074085888 count=20480

dd if=/proc/6821/mem of=/sdcard/mem bs=1skip=1336799232 count=28672

root@android:/	crackme.so
cat /proc/6821/maps ¦ grep libcrackme.so	
4fadf000-4fae0000 r-xp 00000000 b3:13 2361	/data/app-lib/com.yaotong.crack
me-1/libcrackme.so	
4fae0000-4fae1000 rwxp 00001000 b3:13 2361	/data/app-lib/com.yaotong.crack
me-1/libcrackme.so	
4fae1000-4fae3000 r-xp 00002000 b3:13 2361	/data/app-lib/com.yaotong.crack
me-1/libcrackme.so	
4fae3000-4fae4000 rwxp 00004000 b3:13 2361	/data/app-lib/com.yaotong.crack
me-1/libcrackme.so	
4fae4000-4fae5000 rp 00004000 b3:13 2361	/data/app-lib/com.yaotong.crack
me-1/libcrackme.so http://blog.csdn.net/	phoebe_2012
4fae5000-4fae6000 rw-p 00005000 b3:13 2361	/data/app-lib/com.yaotong.crack
me-1/libcrackme.so	
root@android:/	d∕mem bs=1 skip=1336799232 count=
28672	
/mem bs=1 skip=1336799232 count=28672	<
28672+0 records in	
28672+0 records out	
28672 bytes transferred in 1.021 secs (28082 byt	tes/sec)
root@android:/ #	

mem文件用IDA打开,查看0x00004450处的值。

00004450 61 69 79 6F 75 2C 62 75 63 75 6F 6F 00 4C 37 39 aiyou,bucuoo.L79

方法三:

执行android server: adb shell /data/local/tmp/android server

端口转发adb forward tcp:23946 tcp:23946

在模拟器上运行程序

ps找到程序pid

命令: kill -19 <pid> 可以让进程暂停

IDA附加

libcrackme.so基址为0x4914C000

aWojiushidaan: 0x00004450

查看0x49150450处的内存 49150450 61 69 79 6F 75 2C 62 75 67 6F 6F 00 4C 37 39 aiyou,bucuoo.L79

第三题的解决方案:

(1) Indroid(LoCCS实验室开发的工具)貌似要在google手机编译后才能运行成功;

(2) Zjdroid修复一下再内存dump

(3) 过反调试 j_j_ptrace、dump内存、修复dex、修复bakesmali、修复dextojar 或 JEB查看,最后应用 源代码分析

- [1] http://bbs.pediy.com/
- [2] http://www.52pojie.cn/
- [3] http://blog.dutsec.cn/
- [4] http://l-team.org/
- [5] http://le4f.net/

