

[CSDN首页 \(http://www.csdn.net?ref=toolbar\)](http://www.csdn.net?ref=toolbar)[学院 \(http://edu.csdn.net?ref=toolbar\)](http://edu.csdn.net?ref=toolbar)[下载 \(http://download.csdn.net?ref=toolbar\)](http://download.csdn.net?ref=toolbar)

更多 ▼

[下载 CSDN APP \(http://www.csdn.net/app/?ref=toolbar\)](http://www.csdn.net/app/?ref=toolbar)[写博客 \(http://write.blog.csdn.net/postedit?ref=toolbar\)](http://write.blog.csdn.net/postedit?ref=toolbar)[登录 \(https://passport.csdn.net/account/login?ref=toolbar\)](http://blog.csdn.net/) | [注册 \(http://passport.csdn.net/account/mobile/register?ref=toolbar&action=mobileRegister\)](http://passport.csdn.net/account/mobile/register?ref=toolbar&action=mobileRegister)[首页 \(http://blog.csdn.net/\)](http://blog.csdn.net/)[移动开发 \(http://blog.csdn.net/mobile/newarticle.html\)](http://blog.csdn.net/mobile/newarticle.html)

全部 ▼

**CSDN** (<http://www.csdn.net>)

目录

## Android代码调试工具 traceview 和 dmtracedump的波折演绎



喜欢

原创 2011年08月25日 00:48:56

22594

75

10



收藏



评论



分享

# Android 程序调试工具

Google为我们提供的代码调试工具的亮点：**traceview** 和 **dmtracedump**。有了这两个工具，我们调试程序分析bug就非常得心应手了。**traceview**帮助我们分析程序性能，**dmtracedump**生成函数调用图。遗憾的是，google提供的**dmtracedump**是个失败的工具，并不能绘图，本文会详细介绍解决方案，实现绘图。

## 生成.trace文件

`android.os.Debug`类，其中重要的两个方法`Debug.startMethodTracing()`和`Debug.stopMethodTracing()`。这两个方法用来创建.trace文件，将从`Debug.startMethodTracing()`开始，到`Debug.stopMethodTracing()`结束，期间所有的调用过程保存在.trace文件中，包括调用的函数名称和执行



灵芝价格



人工智能培训

码云

未开通

<https://github.com>念茜 (<http://blog.csdn.net/yiyaaixuexi>)

+ 关注

原创

136

粉丝

11912

喜欢

0

### 他的最新文章

[更多文章 \(http://blog.csdn.net/yiyaaixuexi\)](http://blog.csdn.net/yiyaaixuexi)

iOS安全攻防（二十四）：敏感逻辑的保护方案（1）

[\(/yiyaaixuexi/article/details/29210413\)](http://yiyaaixuexi/article/details/29210413)

iOS安全攻防（二十三）：Objective-C 代码混淆

[\(/yiyaaixuexi/article/details/29201699\)](http://yiyaaixuexi/article/details/29201699)

iOS安全攻防（二十二）：static和被裁的符号表

[\(/yiyaaixuexi/article/details/21469769\)](http://yiyaaixuexi/article/details/21469769)

iOS安全攻防（二十）：越狱检测的攻与防

[\(/yiyaaixuexi/article/details/20286929\)](http://yiyaaixuexi/article/details/20286929)

返回顶部

的时间等信息。

把下面代码分别在加在调试起始代码的位置，和终止位置。

```
[java]
1.  Debug.startMethodTracing("test");
2.  Debug.stopMethodTracing();
```



目录



喜欢



收藏



评论



分享

其中参数test是要创建的trace文件的名称，test.trace。默认路径是/sdcard/test.trace，也可以自己制定/data/log/test，表示文件在/data/log/test.trace。

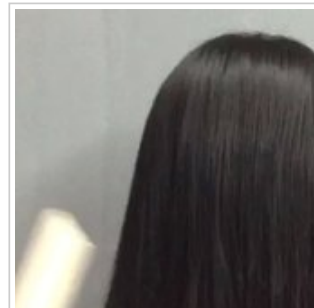
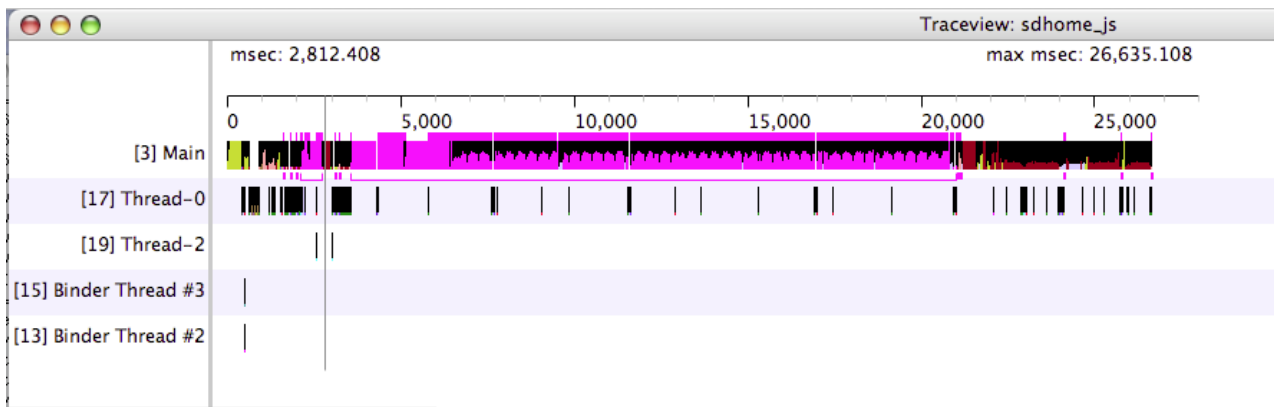
## traceview

在SDK中执行：

./traceview test.trace

我们可以得到

1. 程序中每个线程调用方法的启动和停止时间



种头发危害



灵芝价格



人工智能培训

编辑推荐

最新专栏

Android官方命令深入分析之dmtracedu...

使用dmtracedump生成Android API调用...

dmtracedump 的替代品 (/zujoe/article/...

大学时期要做的50件事 (/davislien/artic...

iOS安全-浅谈关于iOS加固的几种方法...

在线课程



返回顶部

## 2.函数执行的信息和效率分析

Name	Incl %	Inclusive	Excl %	Exclusive	Calls+Rec
4 android/webkit/LoadListener.nativeFinished (JV	66.6%	17734.382	53.2%	14161.950	14+0
3 android/webkit/LoadListener.tearDown (JV	100.0%	17734.382			14/14
6 android/view/View.invalidate (III)V	19.8%	3516.410			2413/2853
57 android/webkit/BrowserFrame.startLoadingResource (Ljava	0.3%	44.636			3/15
53 java/util/HashMap.put (Ljava/lang/Object;Ljava/lang/Objec	0.0%	6.223			6/326
20 android/webkit/JWebCoreJavaBridge.setSharedTimer (JV	0.0%	2.593			2/730
378 android/view/ViewGroup.requestLayout (JV	0.0%	1.139			2/54
315 java/util/HashMap.<init> (JV	0.0%	0.879			3/41
629 android/webkit/BrowserFrame.loadCompleted (JV	0.0%	0.285			1/1
598 android/webkit/WebView.didFirstLayout (JV	0.0%	0.231			1/2
703 android/webkit/BrowserFrame.windowObjectCleared (JV	0.0%	0.036			1/2
5 android/webkit/JWebCoreJavaBridge\$TimerHandler.handleMessa	16.3%	4342.697	0.5%	132.018	730+0
6 android/view/View.invalidate (III)V	15.6%	4161.341	1.2%	319.164	2853+0
7 android/webkit/JWebCoreJavaBridge.access\$300 (Landroid/webk	15.1%	4025.658	0.1%	26.727	729+0
8 android/webkit/JWebCoreJavaBridge.sharedTimerFired (JV	15.0%	3998.931	8.5%	2256.801	729+0
9 android/view/View.invalidate (Landroid/graphics/Rect;)V	13.8%	3671.342	0.9%	246.190	2853+0
10 android/view/ViewGroup.invalidateChild (Landroid/view/View;La	12.4%	3298.987	6.3%	1687.629	876+1148
11 android/event/EventLoop.processPendingEvents (JV	6.3%	1674.317	0.6%	151.201	12+0
12 android/view/ViewRoot.handleMessage (Landroid/os/Message;)V	4.6%	1217.210	0.0%	1.992	35+0
13 android/view/ViewRoot.performTraversals (JV	4.5%	1209.815	0.0%	7.190	34+0
14 android/view/ViewRoot.draw (Z)V	4.1%	1096.832	0.0%	11.508	34+0
15 android/policy/PhoneWindow\$DecorView.drawTraversal (Landrc	3.9%	1040.408	0.0%	2.218	34+0
16 android/widget/FrameLayout.drawTraversal (Landroid/graphics,	3.8%	1023.779	0.0%	3.129	34+48
17 android/view/View.drawTraversal (Landroid/graphics/Canvas;La	3.8%	1022.611	0.1%	19.213	34+154
18 android/view/ViewGroup.dispatchDrawTraversal (Landroid/grap	3.8%	1000.413	0.2%	42.609	34+130
19 android/view/ViewGroup.drawChild (Landroid/graphics/Canvas;	3.7%	983.346	0.2%	42.926	34+150
20 android/webkit/JWebCoreJavaBridge.setSharedTimer (JV	3.5%	929.506	0.2%	57.241	730+0
21 android/webkit/WebView.nativeDrawRect (Landroid/graphics/C	3.5%	923.805	3.0%	807.952	15+0
22 android/net/http/QueuedRequest.start (Landroid/net/http/Que	3.2%	847.172	0.0%	3.556	15+0
23 android/net/http/QueuedRequest\$QREventHandler.endData (JV	3.1%	828.592	0.0%	1.619	15+0
24 android/net/http/QueuedRequest.setupRequest (JV	3.1%	819.888	0.0%	5.860	15+0
25 android/net/http/QueuedRequest.requestComplete (JV	3.1%	816.585	0.0%	1.506	15+0
26 android/webkit/CookieManager.getCookie (Landroid/content/Cc	2.7%	722.837	0.0%	8.081	15+0
27 android/webkit/LoadListener.commitLoad (JV	2.6%	688.168	0.1%	17.708	58+0
28 android/webkit/LoadListener.nativeAddData (B)V	2.3%	621.864	1.2%	306.817	57+0
29 android/graphics/Rect.offset (II)V	2.2%	573.985	2.2%	573.985	17210+0

 dmtracedump



Python即将成为第一语言  
(http://edu.csdn.net/huiyi  
utm\_source=blog9)  
Course/detail/569?



构建企业级高性能OLAP引擎  
(http://edu.csdn.net/huiyi  
utm\_source=blog9)  
Course/detail/532?



灵芝价格



人工智能培训

↑  
返回顶部

dmtracedump原本的用意是将整个调用过程和时间分析结合，以函数调用图的形式表现出来。可是google这个项目一直处于broken状态，迟迟不能得到完善。现在的dmtracedump只有-o选项可以使用，在终端上列出函数调用信息，和traceview功能相似，如果执行./dmtracedump -g test.png test.trace就会卡住不动。

起初我以为是test.trace文件的问题，对文件的结束符稍作修改，画出了一副雷人的图片：

[0] (toplevel).(null) (0, 0, 0)



目录



喜欢



收藏



评论



分享

后来，搜到了网络上有牛人提供了dmtracedump的替代

(<http://blog.csdn.net/zjujoe/article/details/6080738>)（这是原文链接），是一个python脚本，借助dot来绘制矢量图。python脚本一定要注意对齐格式，对齐缩进就是他的逻辑结构。

[python]

```
1.  #!/usr/bin/env python
2.  """
3.  turn the traceview data into a jpg pic, showing methods call relationship
4.  """
5.  import sys
6.  import os
7.  import struct
8.  import re
9.  #####
10. ##### Global Variable #####
11. #####
12. target_thread=1 #the thread that we want to track, filt out other threads
13. #all_actions = ["enter","exit","exception","reserved"]
14. all_threads = {}
15. all_methods = {}
16. all_records = []
17. parent_methods = {}
18. child_methods = {}
19. method_calls = {}
20. #####
21. ##### Methods #####
22. #####
23. def add_one_thread(line):
```



返回顶部



目录



喜欢



收藏



评论



分享

```
24.     fields = line.split("/t")
25.     all_threads[int(fields[0],10)]=fields
26. def add_one_method(line):
27.     fields = line.split("/t")
28.     all_methods[int(fields[0],16)]=fields
29. def add_one_record(one):
30.     thread_id,=struct.unpack("B",one[:1])
31.     if (thread_id == target_thread):
32.         tmp,=struct.unpack("L",one[1:5])
33.         method_id= (tmp / 4) * 4;
34.         method_action= tmp % 4;
35.         time_offset,=struct.unpack("L",one[5:])
36.         all_records.append([thread_id, method_id, method_action, time_offset])
37. def handle_one_call(parent_method_id,method_id):
38.     if not (parent_methods.has_key(parent_method_id)):
39.         parent_methods[parent_method_id]=1
40.     if not (child_methods.has_key(method_id)):
41.         child_methods[method_id]=1
42.     if method_calls.has_key(parent_method_id):
43.         if method_calls[parent_method_id].has_key(method_id):
44.             method_calls[parent_method_id][method_id]+=1
45.         else:
46.             method_calls[parent_method_id][method_id]=1
47.     else:
48.         method_calls[parent_method_id]={}
49.         method_calls[parent_method_id][method_id]=1
50. def gen_funcname(method_id):
51.     r1=re.compile(r'[/\{1}lt;>')
52.     str1=r1.sub("_",all_methods[method_id][1])
53.     str2=r1.sub("_",all_methods[method_id][2])
54.     return str1+"_"+str2
55. def gen_dot_script_file():
56.     myoutfile = open("graph.dot", "w")
57.     myoutfile.write("digraph vanzo {\n/n/n");
58.     for one in all_methods.keys():
59.         if parent_methods.has_key(one):
60.             myoutfile.write(gen_funcname(one)+"    [shape=rectangle];/n")
61.         else:
62.             if child_methods.has_key(one):
63.                 myoutfile.write(gen_funcname(one)+"    [shape=ellipse];/n")
64.     for one in method_calls.keys():
```



灵芝价格



人工智能培训



返回顶部



- 目录
- 喜欢
- 收藏
- 评论
- 分享

```
65.         for two in method_calls[one]:
66.             myoutfile.write(gen_funcname(one) + ' -
> ' + gen_funcname(two) + ' [label="' + str(method_calls[one]
[two]) + '" fontsize="10"];/n')
67.             myoutfile.write("/n}/n");
68.             myoutfile.close
69.
70. #####
71. ##### Script starts from here #####
72. #####
73. if len(sys.argv) < 2:
74.     print 'No input file specified.'
75.     sys.exit()
76. if not (os.path.exists(sys.argv[1])):
77.     print "input file not exists"
78.     sys.exit()
79. #Now handle the text part
80. current_section=0
81. for line in open(sys.argv[1]):
82.     line2 = line.strip()
83.     if (line2.startswith("*")):
84.         if (line2.startswith("*version")):
85.             current_section=1
86.         else:
87.             if (line2.startswith("*threads")):
88.                 current_section=2
89.             else:
90.                 if (line2.startswith("*methods")):
91.                     current_section=3
92.                 else:
93.                     if (line2.startswith("*end")):
94.                         current_section=4
95.                     break
96.             continue
97.         if current_section==2:
98.             add_one_thread(line2)
99.         if current_section==3:
100.             add_one_method(line2)
101.
102. #Now handle the binary part
103. mybinfile = open(sys.argv[1], "rb")
```





目录



喜欢



收藏



评论



分享

```

104. alldata = mybinfile.read()
105. mybinfile.close()
106. pos=alldata.find("SLOW")
107. offset,=struct.unpack("H",alldata[pos+6:pos+8])
108. pos2=pos+offset #pos2 is where the record begin
109. numofrecords = len(alldata) - pos2
110. numofrecords = numofrecords / 9
111. for i in xrange(numofrecords):
112.     add_one_record(alldata[pos2 + i * 9:pos2 + i * 9 + 9])
113. my_stack=[0]
114. for onerecord in all_records:
115.     thread_id=onerecord[0];
116.     method_id=onerecord[1];
117.     action=onerecord[2];
118.     time=onerecord[3];
119.     if(action==0):
120.         if(len(my_stack) > 1):
121.             parent_method_id=my_stack[-1]
122.             handle_one_call(parent_method_id,method_id)
123.             my_stack.append(method_id)
124.     else:
125.         if(action==1):
126.             if(len(my_stack) > 1):
127.                 my_stack.pop()
128. gen_dot_script_file()
129. os.system("dot -Tjpg graph.dot -o output.jpg;rm -f graph.dot");

```

修改，/t变为\t，/n变为\n。

在计算器的源码onCreate中添加

[java]

```

1. Debug.startMethodTracing("calc");
2. Log.v(LOG_TAGS"+++++++test+++++++");

```



灵芝价格



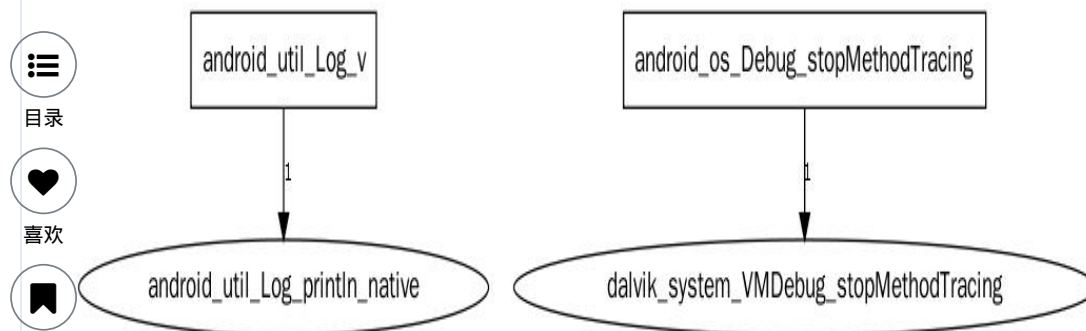
人工智能培训



返回顶部

3. `Debug.stopMethodTracing();`

运行脚本得到calc.trace，画出out.jpg



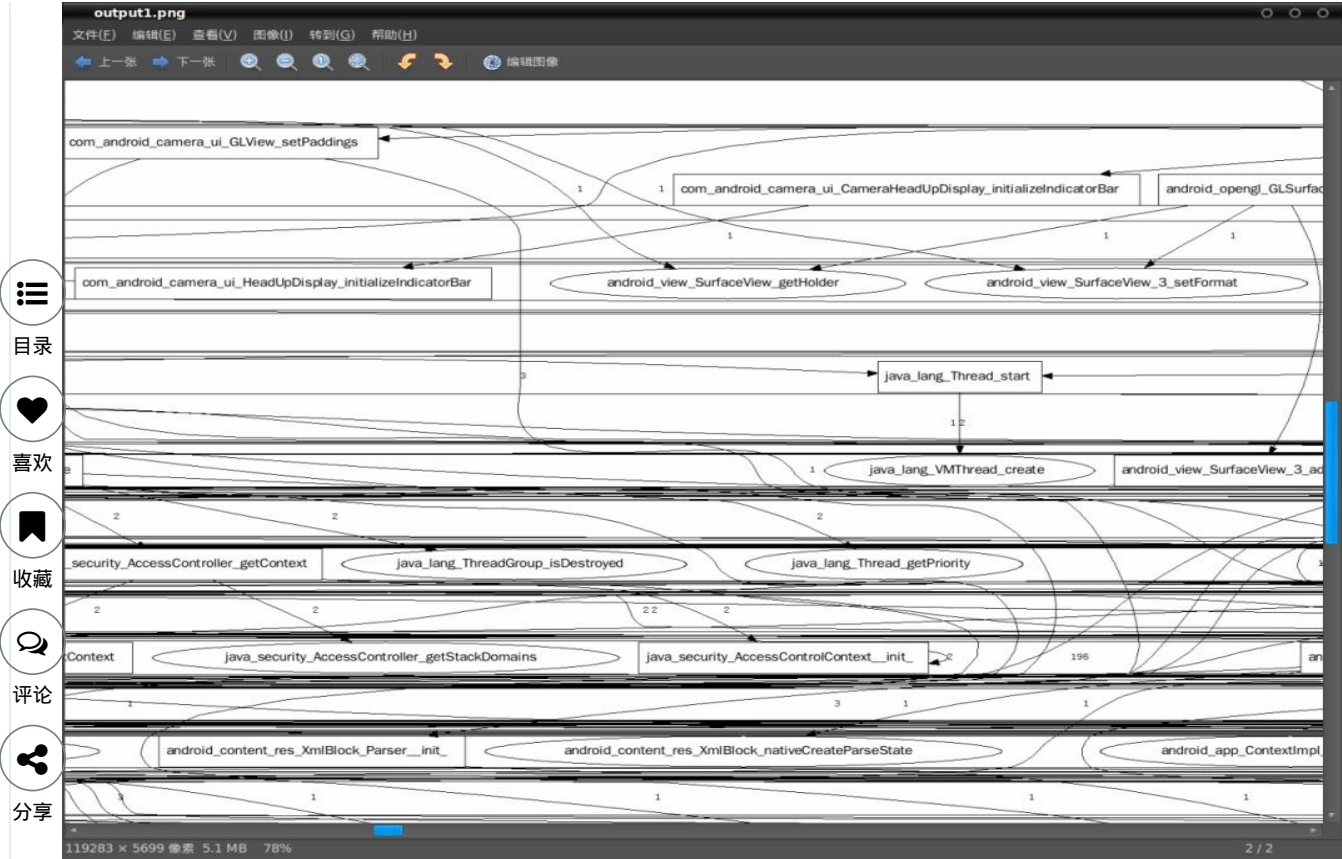
可当得到的trace文件比较复杂时画图就会崩溃。修改脚本最后执行dot命令时的参数，jpg格式太大容易

崩溃，-Tjpg改为-Tpng:gd，画大幅png。

我拿camera做了个实验，得到的png甚大，这是其中一角：







OK累死我了，又快1点了！>\_< 晚安！🌙



灵芝价格



人工智能培训

↑  
返回顶部

版权声明：本文为博主原创文章，未经博主允许不得转载。

举报

标签：工具 (http://so.csdn.net/so/search/s.do?q=工具&t=blog) /

android (http://so.csdn.net/so/search/s.do?q=android&t=blog) /

methods (http://so.csdn.net/so/search/s.do?q=methods&t=blog) /

thread (http://so.csdn.net/so/search/s.do?q=thread&t=blog) /

程序调试工具 (http://so.csdn.net/so/search/s.do?q=程序调试工具&t=blog) /

目录action (http://so.csdn.net/so/search/s.do?q=action&t=blog)

喜欢

收藏

评论

分享

回复

46楼

查看 72 条热评

相关文章推荐

Android官方命令深入分析之dmtracedump (/song19891121/article/details/47149605)

dmtracedump是一个根据log文件生成图形化调用堆栈的工具（除了Traceview之外）。dmtracedump的用法：dmtracedump [-ho] [-s sortable] [-d ...



 song19891121 (<http://blog.csdn.net/song19891121>) 2015-07-30 14:34 954

## 使用dmtracedump生成Android API调用图 (/cpcpcp123/article/details/53809038)

在Android的性能分析中，dmtracedump工具可以从trace文件中生成关系图。dmtracedump使用了graphviz来生成图片，所以需要安装graphviz。一：首先生成...



 cpcpcp123 (<http://blog.csdn.net/cpcpcp123>) 2016-12-22 09:33 769



喜欢



收藏



## 我是如何成为一名python大咖的？ ([http://edu.csdn.net/topic/python1?utm\\_source=blog10](http://edu.csdn.net/topic/python1?utm_source=blog10))

人生苦短，都说必须python，那么我分享下我是如何从小白成为Python资深开发者的吧。2014年我大学刚毕业..




评论



分享

## dmtracedump 的替代品 (/zjujoe/article/details/6080738)

这两天使用 traceview 分析代码性能，google文档上说有一个工具 dmtracedump，可以生成函数调用图。试验了一下，失败！google 一下，原来此工具一直处于 br...

 zjujoe (<http://blog.csdn.net/zjujoe>) 2010-12-16 19:00 6573

## 大学时期要做的50件事 (/davislien/article/details/51339648)

大学时期要做的50件事 (2012-06-01 21:48:56) 转载▼ 标签： 大学 杂谈 分类： 大学生生活 ...

 DavisLien (<http://blog.csdn.net/DavisLien>) 2016-05-07 19:49 12014



灵芝价格




人工智能培训



返回顶部


## iOS安全-浅谈关于iOS加固的几种方法，代码混淆，类名方法名混淆等 (/demondev/article/details/52136342)



转自：<http://www.blogfshare.com/ios-protect.html> 关于IOS安全这方面呢，能做的安全保护确实要比Android平台下面能做的少很多。...

 demondev (<http://blog.csdn.net/demondev>) 2016-08-06 15:01 4721





### 目录 iOS应用安全之代码混淆设计篇 (/zm53373581/article/details/49053269)

 iOS应用安全随着各种事件的曝出，越来越受到重视。那针对iOS应用安全方面能做点什么呢？如何让我们开发的应用更安全喜欢一点呢？要知道如何才能安全，就要了解iOS应用怎么就不安全了呢？现在随着越狱技术的提高...

  zm53373581 (<http://blog.csdn.net/zm53373581>) 2015-10-11 21:35 4989  
收藏

### 评论 iOS应用安全之代码混淆实现篇 (/zm53373581/article/details/49120895)

评论  
iOS应用安全之代码混淆实现篇

  zm53373581 (<http://blog.csdn.net/zm53373581>) 2015-10-14 09:40 9859  
分享

## [iOS 代码混要加固转帖] (/iostiannan/article/details/72868598)

iOS安全-浅谈关于iOS加固的几种方法，代码混淆，类名方法名混淆等 <http://blog.csdn.net/demondev/article/details/52136342>  
2iOS 对源代码进行...



 iOSTianNan (<http://blog.csdn.net/iOSTianNan>) 2017-06-05 16:31 337

## [ios]安全攻防之代码混淆的一个小工具 (/yxh265/article/details/38438959)


看了“念茜”的这 <http://blog.csdn.net/yiyaaixuexi/article/details/29201699> #include #include #include...




返回顶部


 yxh265 (<http://blog.csdn.net/yxh265>) 2014-08-08 14:38  7346



**Android代码调试工具 traceview 和 dmtracedump的波折演绎 .**  
**(/yangchaofeng1229/article/details/6730603)**


 Android 程序调试工具 Google为我们提供的代码调试工具的亮点：traceview 和 dmtracedump。有了这两个工具，我们调试程序分析bug就非常得心应手了。t...


 目录  yangchaofeng1229 (<http://blog.csdn.net/yangchaofeng1229>) 2011-08-29 20:57  516

 喜欢 **Android代码调试工具 traceview 和 dmtracedump的波折演绎**  
**(/huigezhang/article/details/6723562)**

 Android 程序调试工具 Google为我们提供的代码调试工具的亮点：traceview 和 dmtracedump。有了这两个工具，我们调试程序分析bug就非常得心应手了。tr...

 评论  HuiGeZhang (<http://blog.csdn.net/HuiGeZhang>) 2011-08-26 22:16  982

 分享 **Android代码调试工具 traceview 和 dmtracedump的波折演绎**  
**(/zhjp4295216/article/details/6728386)**

 Android 程序调试工具 Google为我们提供的代码调试工具的亮点：traceview 和 dmtracedump。有了这两个工具，我们调试程序分析bug就非常得心应手了。tr...


 zhjp4295216 (<http://blog.csdn.net/zhjp4295216>) 2011-08-29 10:31  460

**Android代码调试工具 traceview 和 dmtracedump的波折演绎 .**  
**(/caoliang0921/article/details/18098967)**



  
返回顶部

[+] 该文转自<http://blog.csdn.net/yiyaaixuexi/article/details/6716884> ♡Android 程序调试工具 Google为我们提供...

 caoliang0921 (<http://blog.csdn.net/caoliang0921>) 2014-01-10 22:46 509


## Android代码调试工具 traceview 和 dmtracedump的波折演绎 (/hupuxiang/article/details/7317589)



目录 ♡Android 程序调试工具 Google为我们提供的代码调试工具的亮点：traceview 和 dmtracedump。有了这两个工具，我们调试程序分析bug就非常得心应手了。tr...



喜欢

 hupuxiang (<http://blog.csdn.net/hupuxiang>) 2012-03-04 12:13 365




收藏

## Android程序调试工具TraceView和dmtracedump (/tongyan7952/article/details/7663189)

这两个工具是android sdk自带工具，在android-sdk-windows\tools中可以找到；1.traceview分析程序性能 Debug.startMethodTracing



评论


 tongyan7952 (<http://blog.csdn.net/tongyan7952>) 2012-06-14 15:29 416



分享

## Android调试工具--Traceview (/l892799475/article/details/44055895)

作为优秀的android应用程序开发者，调试程序是一个重要技能。现看例子吧：Traceview是android平台配备的一个很好的性能分析工具。它可以通过图形化的方式让我们了解我们要跟...

 l892799475 (<http://blog.csdn.net/l892799475>) 2015-03-04 09:35 198

## Android调试工具之Traceview (/huanglong1987/article/details/8806716)

Android调试工具之Traceview Traceview是android平台配备的一个很好的性能分析工具。它可以通过图形化的方式让我们了解我们要跟踪的程序的性能，并且能具体到method。...




灵芝价格

人工智能培训





返回顶部



 Huanglong1987 (<http://blog.csdn.net/Huanglong1987>) 2013-04-16 00:04 471

**Android调试工具之Traceview (/ljobtong/article/details/7790824)**


Traceview是android平台配备的一个很好的性能分析工具。它可以通过图形化的方式让我们了解我们要跟踪的程序的性能，并且能具体到method。 进行Traceview的版本限制 对于And...

  shinilaobababa (<http://blog.csdn.net/shinilaobababa>) 2012-07-26 20:00 442


目录

 **Traceview调试工具详解 (/mynote/article/details/39779655)**

喜欢

 Android调试工具之Traceview Traceview是android平台配备的一个很好的性能分析工具。它可以通过图形化的方式让我们了解我们要跟踪的程序的性能，并且能具体到met...

收藏

 mynote (<http://blog.csdn.net/mynote>) 2014-10-04 14:45 615

评论

评论



分享



  
返回顶部