

蓝屏 **dump** 分析教程，附分析工具 **WinDbg(x86 x64)6.12.0002.633** 下载

时间:2012年09月14日 | 栏目:技术方案 | 评论:24条评论 | 点击:131,152次+复制本文链接

本文标签: 工具 , 蓝屏

一、**WinDbg** 是什么？它能做什么？

WinDbg 是在 windows 平台下，强大的用户态和内核态调试工具。它能够通过 **dmp** 文件轻松的定位到问题根源，可用于分析蓝屏、程序崩溃（**IE** 崩溃）原因，是我们日常工作中必不可少的一个有力工具，学会使用它，将有效提升我们的问题解决效率和准确率。

二、**WinDbg6.12.0002.633** 下载：

x86 位版本下载：【微软官方安装版】

蓝屏 Dump 分析工具 WinDbg(x86).rar 31,204 次

x64 位版本下载：【微软官方安装版】

蓝屏 Dump 分析工具 WinDbg(x64).rar 30,900 次

三、设置符号表：

符号表是 **WinDbg** 关键的 数据库 ，如果没有它，**WinDbg** 基本上就是个废物，无法分析出更多问题原因。所以使用 **WinDbg** 设置符号表，是必须要走的一步。

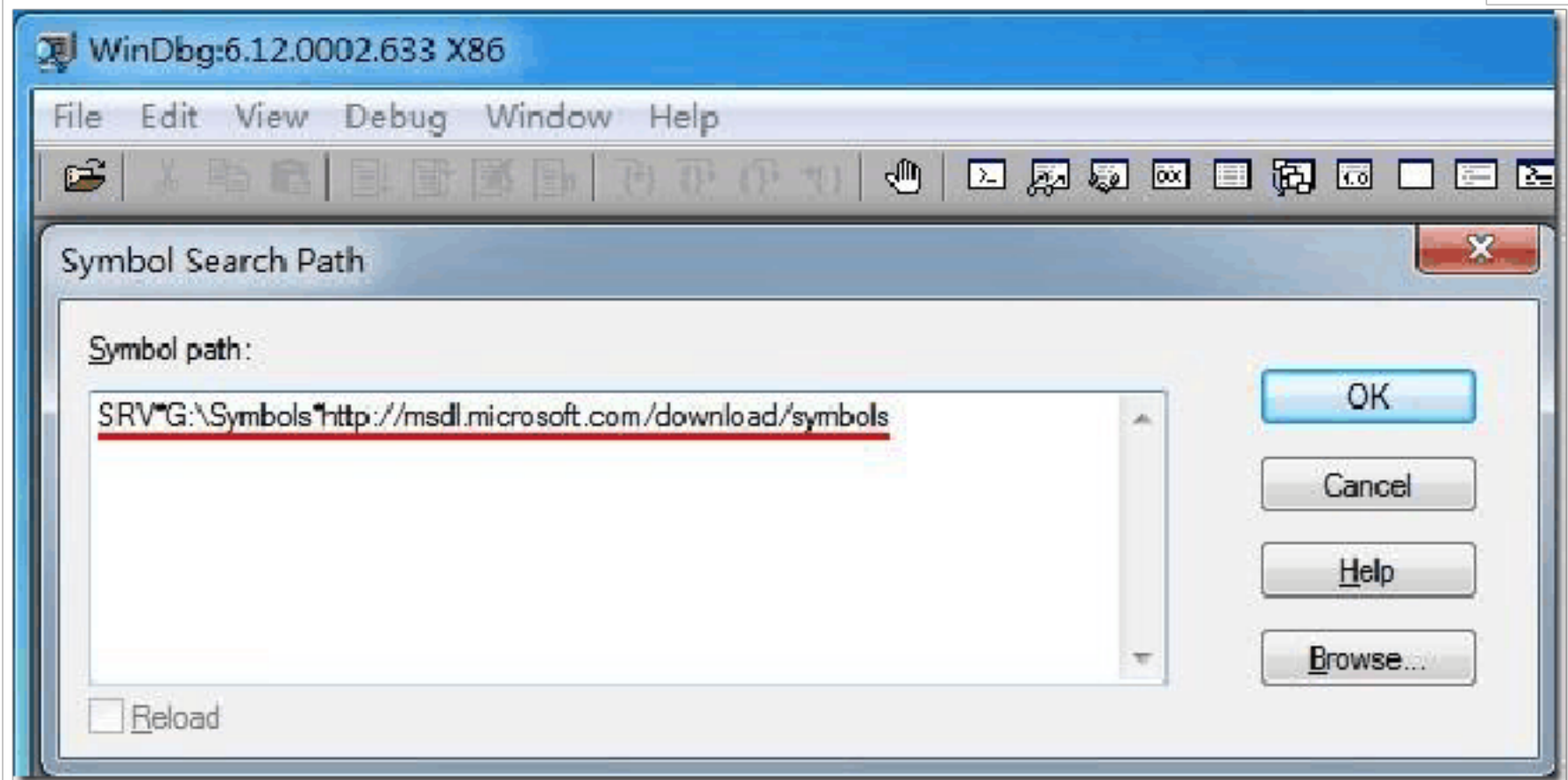
1、运行 **WinDbg** 软件，然后按【**Ctrl+S**】弹出符号表设置窗

2、将符号表地址：

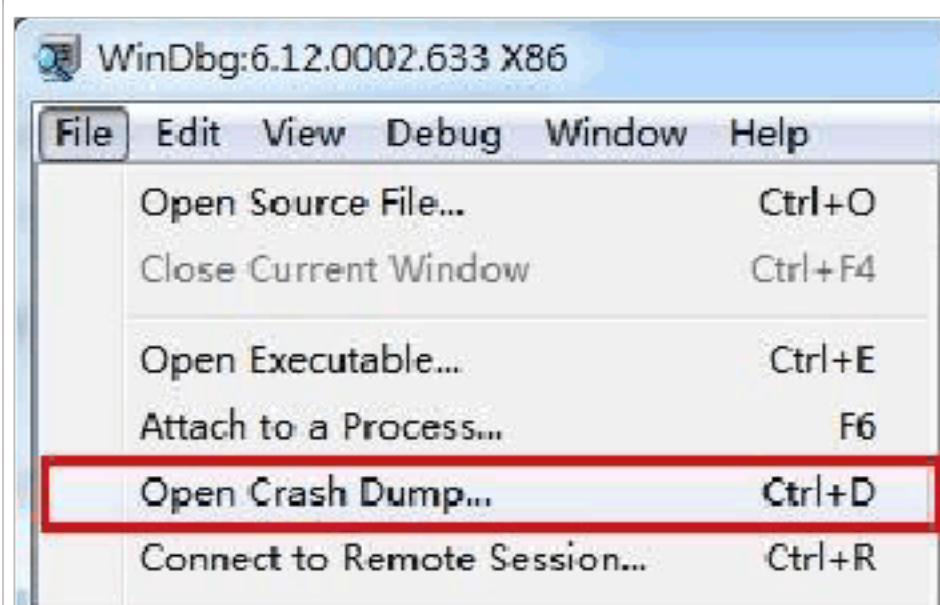
SRV*C:\Symbols*http://msdl.microsoft.com/download/symbols 贴在输入框

中，点击确定即可。

注：红色字体为符号表本地存储路径，建议固定路径，可避免符号表重复下载。

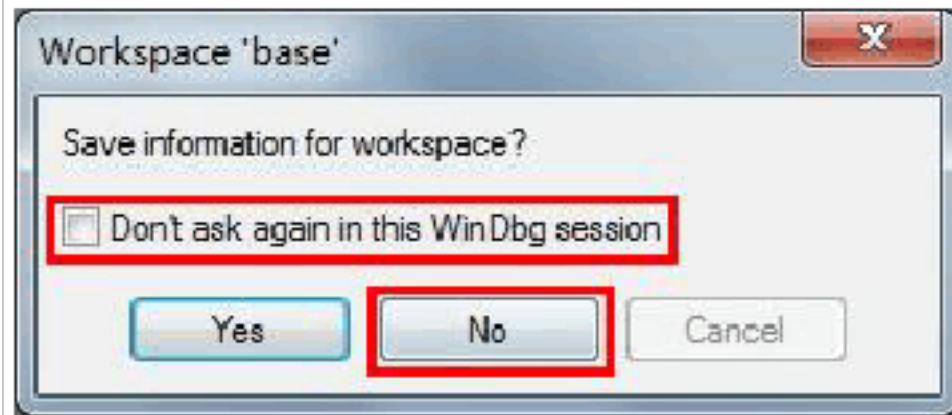


四、学会打开第一个 **dmp** 文件！



当你拿到一个 dmp 文件后，可使用 **【Ctrl+D】** 快捷键来打开一个 dmp 文件，或者点击 WinDbg 界面上的 **【File=>Open Crash Dump...】** 按钮，来打开一个 dmp 文件。第一次

打开 dmp 文件时，可能会收到如下提示，出现这个提示时，勾选 **Don't ask again in this WinDbg session** 然后点否即可。



当你想打开第二个 dmp 文件时，可能因为上一个分析记录未清除，导致无法直接分析下一个 dmp 文件，此时你可以使用快捷键 **【Shift+F5】** 来关闭上一个 dmp 分析记录。

至此，简单的 WinDbg 使用你已经学会了！

五、通过简单的几个步骤学会分析一些 dmp 文件。

分享一个 **SE** 蓝屏 dmp 案例的分析过程：

当你打开一个 dmp 文件后，可能因为太多信息，让你无所适从，不过没关系，我们只需要关注几个关键信息即可。

第一个关键信息：**System Uptime**（开机时间）：

通过观察这个时间你就可以知道问题是在什么时候出现的，例如时间小于 1 分钟基本可以定位为开机蓝屏，反之大于一分钟则可证明是上机后或玩的过程中出现问题了。

接下来用一个简单的例子来学习简单的 dmp 分析，下图中 **System Uptime: 0 days**

0:14:23.581，意思是 0 天(days)0 小时 14 分 23 秒 581 毫秒时出现蓝屏了，看来是上机没多久就蓝屏了，这位顾客很悲催

```

Microsoft (R) Windows Debugger Version 6.12.0002.633 X86
Copyright (c) Microsoft Corporation. All rights reserved.

Loading Dump File [g:\Backup\Filebackup\桌面\黑龙江\红兴 dump\0000008E_192.168.0.138]
Mini Kernel Dump File: Only registers and stack trace are available

Symbol search path is: SRV*g:\symbols*http://msdl.microsoft.com/download/symbols
Executable search path is:
Windows XP Kernel Version 2600 (Service Pack 3) MP (2 procs) Free x86 compatible
Product: WinNt, suite: TerminalServer SingleUserTS
Built by: 2600.xpsp_sp3_qfe.111025-1623
Machine Name:
Kernel base = 0x804e4000 PsLoadedModuleList = 0x8056a720
Debug session time: Sat Jul 14 00:20:30.812 2012 (UTC + 8:00)
System Uptime: 0 days 0:14:23.581
Loading Kernel Symbols
.....
Loading User Symbols
Mini Kernel Dump does not contain unloaded driver list
*****
*
*           Bugcheck Analysis
*
*****

Use !analyze -v to get detailed debugging information.

BugCheck 8E, {c0000005, 8053b7bb, b2bfeb78, 0}

Unable to load image KiMsgProtect.sys, Win32 error 0n2
*** WARNING: Unable to verify timestamp for KiMsgProtect.sys
*** ERROR: Module load completed but symbols could not be loaded for KiMsgProtect.s
Probably caused by : KiMsgProtect.sys ( KiMsgProtect+1496 )

Followup: MachineOwner
-----

```

那么是什么导致蓝屏的呢？接下来我们就要注意第二个关键信息了！

第二个关键信息：**Probaly caused by**（造成蓝屏可能的原因）

这个信息是相对比较重要的一个信息，如果你运气好的话，通过这个信息基本上可以看到导致蓝屏的驱动或者程序名称了，就像下图一样，初步的分析已经有了结果，**Probaly caused by** 后面显示的是一个名为 **KiMsgProtect.sys** 的驱动文件导致蓝屏，这个文件就是恒信一卡通的一个关键驱动。因此蓝屏则很有可能和一卡通有关。

括号中驱动文件名后面的+号代表的是偏移地址，假如多个 dmp 文件的驱动文件名一样，且偏移地址也一样，则问题原因极有可能是同一个，这个偏移地址与汇编有关，这里不多做介绍。

```
Microsoft (R) Windows Debugger Version 6.12.0002.633 X86
Copyright (c) Microsoft Corporation. All rights reserved.

Loading Dump File [g:\Backup\Filebackup\桌面\黑龙江\红兴 dump\0000008E_192.168.0.138]
Mini Kernel Dump File: Only registers and stack trace are available

Symbol search path is: SRV*g:\symbols*http://msdl.microsoft.com/download/symbols
Executable search path is:
Windows XP Kernel Version 2600 (Service Pack 3) MP (2 procs) Free x86 compatible
Product: WinNt, suite: TerminalServer SingleUserTS
Built by: 2600.xpsp_sp3_qfe.111025-1623
Machine Name:
Kernel base = 0x804e4000 PsLoadedModuleList = 0x8056a720
Debug session time: Sat Jul 14 00:20:30.812 2012 (UTC + 8:00)
System Uptime: 0 days 0:14:23.581
Loading Kernel Symbols
.....
Loading User Symbols
Mini Kernel Dump does not contain unloaded driver list
*****
*
*           Bugcheck Analysis
*
*****

Use !analyze -v to get detailed debugging information.

BugCheck 8E, {c0000005, 8053b7bb, b2bfeb78, 0}

Unable to load image KiMsgProtect.sys, Win32 error 0n2
*** WARNING: Unable to verify timestamp for KiMsgProtect.sys
*** ERROR: Module load completed but symbols could not be loaded for KiMsgProtect.s
Probably caused by : KiMsgProtect.sys ( KiMsgProtect+1496 )

Followup: MachineOwner
-----
```

其实，对于分析蓝屏 dmp 并不是每次运气都那么好，假如刚刚打开 dmp 文件未看到明确的蓝屏原因时，我们就需要借助一个命令来进一步分析 dmp，这个命令就是：**!analyze -v**，这个命令能够自动分析绝大部分蓝屏原因。当初步分析没有结果时，可以使用该命令进一步分析故障原因，当然你也可以直接点击链接样式的**!analyze -v**来进行执行该命令，为了让大家更直观的看懂里面的信息，大家可以直接看图片中的注释信息。

```
0: kd> !analyze -v
```

```
*****
*
*                               Bugcheck Analysis
*
*
*****
```

KERNEL MODE EXCEPTION NOT HANDLED (8e) 这里提示的是蓝屏代码，这个例子中蓝屏代码是8E

This is a very common bugcheck. Usually the exception address pinpoints the driver/function that caused the problem. Always note this address as well as the link date of the driver/image that contains this address. Some common problems are exception code 0x80000003. This means a hard coded breakpoint or assertion was hit, but this system was booted /NODEBUG. This is not supposed to happen as developers should never have hardcoded breakpoints in retail code, but ...
If this happens, make sure a debugger gets connected, and the system is booted /DEBUG. This will let us see why this breakpoint is happening.

这里是WinD
可以使用Go
词典，会对

Arguments:

Arg1: c0000005, The exception code that was not handled
Arg2: 8053b7bb, The address that the exception occurred at
Arg3: b2bfeb78, Trap Frame
Arg4: 00000000

这里是一些蓝屏参数，平时我们看
面会有一个括号，这就是蓝屏界面
字来说，看这有点像看天数，之
了，这里的参数会根据蓝屏代码不
会发现这个现象了……

Debugging Details:

EXCEPTION_CODE: (NTSTATUS) 0xc0000005 - 0x%08lx

FAULTING_IP:

nt!RtlInitUnicodeString+1b
8053b7bb f266af repne scas word ptr es:[edi]

TRAP_FRAME: b2bfeb78 -- (.trap 0xfffffff2b2bfeb78)

ErrCode = 00000000

eax=00000000 ebx=e302c8b0 ecx=fffffff edx=b2bfebfc esi=80570a48 edi=000005b8
eip=8053b7bb esp=b2bfebec ebp=b2bfec08 iopl=0 nv up ei pl zr na pe nc
cs=0008 ss=0010 ds=0023 es=0023 fs=0030 gs=0000 efl=00010246

nt!RtlInitUnicodeString+0x1b:

8053b7bb f266af repne scas word ptr es:[edi]

Resetting default scope

CUSTOMER_CRASH_COUNT: 58

DEFAULT_BUCKET_ID: COMMON_SYSTEM_FAULT

BUGCHECK_STR: 0x8E 这个也是蓝屏代码，通常MSDN上的蓝屏代码表示方法通常是0x8E这样，而不是0x0000008E

PROCESS_NAME: PinyinUp.exe 这里是触发蓝屏的应用程序，可能是.exe可能是.dat也可能是.dll，并不固定，但要注意
因为用户态程序是不会导致蓝屏的，蓝屏只可能是内核态程序(驱动程序)才会导致蓝

LAST_CONTROL_TRANSFER: from 880c56d2 to 8a6d193c

DEFAULT_BUCKET_ID: COMMON_SYSTEM_FAULT

BUGCHECK_STR: 0x8E

PROCESS_NAME: PinyinUp.exe

LAST_CONTROL_TRANSFER: from 880c56d2 to 8a6d193d

STACK_TEXT:

WARNING: Frame IP not in any known module. Following frames may be wrong.
b2bfe708 880c56d2 b2bfe71c b2bfe728 00000000 0x8a6d193c
b2bfe720 80506f43 0000008e c0000005 8053b7bb 0x880c56d2
b2bfe740 8050b827 0000008e c0000005 8053b7bb nt!KeBugCheckEx+0x1b
b2bfeb08 8054f0e5 b2bfeb24 00000000 b2bfeb78 nt!KiDispatchException+0x3b1
b2bfeb70 8054f096 b2bfec08 8053b7bb badb0d00 nt!CommonDispatchException+0x4d
b2bfeb9c 8053d52f b39213dc 00000000 00000023 nt!KiExceptionExit+0x18a
b2bfec08 b83c1496 87aea308 87a8562c 00000200 nt!ElapsedDaysToYears+0xb1
b2bfec28 805deb96 00000b6c 00000c7c 00000000 KiMsgProtect+0x1496
b2bfec4c 805df6a8 00000001 00000006 877ca0e0 nt!PspExitProcess+0x5e
b2bfecf0 805df85d 00000000 b2bfed4c 8050c94f nt!PspExitThread+0x5ae
b2bfecfc 8050c94f 877ca0e0 b2bfed48 b2bfed3c nt!PsExitSpecialApc+0x23
b2bfed4c 8054ef46 00000001 00000000 b2bfed64 nt!KiDeliverApc+0x1af
b2bfed4c 7c93b05c 00000001 00000000 b2bfed64 nt!KiExceptionExit+0x3a
0012fd1c 00000000 00000000 00000000 00000000 0x7c93b05c

STACK_COMMAND: kb

FOLLOWUP_IP:

KiMsgProtect+1496
b83c1496 ?? ???

SYMBOL_STACK_INDEX: 7

SYMBOL_NAME: KiMsgProtect+1496

FOLLOWUP_NAME: MachineOwner

MODULE_NAME: KiMsgProtect

IMAGE_NAME: KiMsgProtect.sys

DEBUG_FLR_IMAGE_TIMESTAMP: 4ecf107f

FAILURE_BUCKET_ID: 0x8E_KiMsgProtect+1496

BUCKET_ID: 0x8E_KiMsgProtect+1496


Followup: MachineOwner

这里的作
致蓝屏
很难理解
很好的理
在红方框
KiMsgPr
其实就是
使用分
通过观察

看了这么多信息之后，这个蓝屏 dmp 到底是怎么回事呢？根据 dmp 给出的信息，应该是：顾客上机 **0 天(days)0 小时 14 分 23 秒 581 毫秒**时，一个名为 **PinyinUp.exe** 触发了 **KiMsgProtect.sys** 这个驱动的一个 **Bug**，导致蓝屏。

那么 PinyinUp.exe 和 KiMsgProtect.sys 都是哪个厂商的？一般要知道这个信息，只能去用户的机器上找了，我去找了之后发现 PinyinUp.exe 是搜狗输入法的自动升级程序，KiMsgProtect.sys 是恒信一卡通这个计费软件的驱动，所以这个 dmp 表示出来的意思看上去是搜狗拼音和恒信一卡通搞在一起，出了问题！当然排除方法很简单，把搜狗输入法的自动升级程序删除掉，再看看是否仍然有蓝屏问题发生就 ok 了！

学到这里，基本上已经可以分析绝大部分 dmp 文件了，但是分析蓝屏 dmp 要比较谨慎，对信息需要重新验证一次才更加保险，验证方法很简单，在 WinDbg 的命令输入框内，输入 **!process** 命令，就可以验证触发蓝屏的程序到底是否正确了。



```
Dump g:\Backup\Filebackup\桌面\黑龙江\红兴dump\0000008E_192.168.0.138_2012_7_14_0_18_58.dmp - WinD
File Edit View Debug Window Help
BUCKET_ID: 0x8E_KiMsgProtect+1496
Followup: MachineOwner
0: kd> !process 这里就是命令输入框了
Ln 0, Col 0 Sys 0:g:\Back Proc 000:0 T
```

运行 **!process** 命令后得到的信息：


```

0: kd> !process
GetPointerFromAddress: unable to read from 8056f134
PROCESS 87aee308 SessionId: none Cid: 0c7c Peb: 7ffdd000 ParentCid: 0b6c
DirBase: 0ad10f80 ObjectTable: e11162f8 HandleCount: <Data Not Accessible>
Image: PinyinUp.exe 确认无误，确实是搜狗输入法升级程序触发了恒信一卡通驱动蓝屏……
VadRoot 8826c720 Vads 9 Clone 0 Private 9 Modified 0 Locked 0
DeviceMap e1b89008
Token e326f2e0
ReadMemory error: Cannot get nt!KeMaximumIncrement value.
ffff0000: Unable to get shared data
ElapsedTime 00:00:00.000
UserTime 00:00:00.000
KernelTime 00:00:00.000
QuotaPoolUsage[PagedPool] 10220
QuotaPoolUsage[NonPagedPool] 360
Working Set Sizes (now,min,max) (25, 50, 345) (100KB, 200KB, 1380KB)
PeakWorkingSetSize 26
VirtualSize 3 Mb
PeakVirtualSize 3 Mb
PageFaultCount 19
MemoryPriority BACKGROUND
BasePriority 8
CommitCharge 29

THREAD 87705c78 Cid 0c7c.0708 Teb: 00000000 Win32Thread: 00000000 RUNNING

```

至此，掌握以上几个简单的分析方法之后，基本上绝大多数 dmp 大家都可以独立分析了，当然 WinDbg 是个强大的工具，同时蓝屏的原因也有很多，如果想分析的足够准确，那么就只有多学多练，多去分析，因为 WinDbg 分析除了懂得几个命令之外，经验更加重要！

合理再给大家一些分析建议：

并不一定每个 dmp 文件都可以分析出有用的结论，因此分析 dmp 并不需要对每个 dmp 文件的结果过分纠结，其实蓝屏 dmp 分析也是观察一个规律或者规模的问题定位方法而已。例如你分析了 10 个 dmp，有 5 个 dmp 都指向同一个蓝屏原因，另外 5 个 dmp 的信息五花八门时，那么你可以完全先处理掉 5 次蓝屏，同一个原因的问题，因为解决了这个问题之后，后面的问题可能就都解决了！

vDiskBus+da6c 这个蓝屏信息是指网维大师蓝屏硬盘的 dmp 捕捉机制，这并不是蓝屏原因，有很多朋友因为文章看到一半就去折腾，结果得出一些错误结论，所以这里特意提醒

下大家，看到 vDiskBus+da6c 这个信息之后，就不要再判断错误了，这个信息可以证实的信息是：这个 **dmp** 文件是通过网维大师蓝屏鹰眼捕捉到的，且是在网维无盘客户机上捕捉到的，其它的就不能代表什么了。