Manually Crashing the Windows during hangs

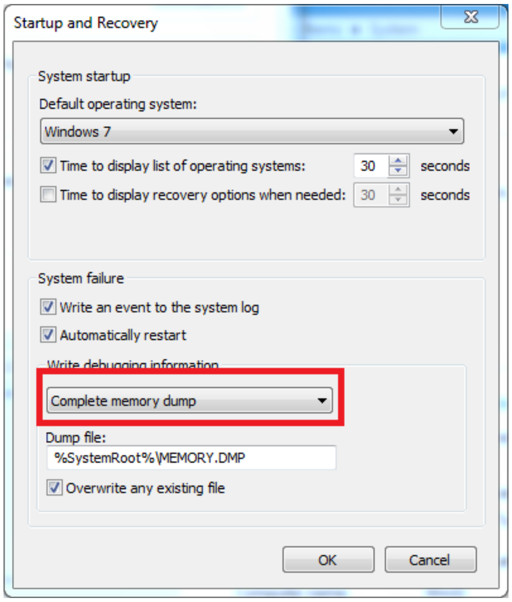
author: Alexandre Borges
date: JUNE/23/2017

Hello readers, how are you? Two days ago, I experienced a known situation: during a presentation about malware analysis, the malware caused a hang of my system (Win7 x86) for few minutes. Of course, as I was using a virtual machine (VMware), I could have suspended the environment and analyzed the **.vmem file**. However, I forced (through Windows) generating a manual crash dump for analyzing it using WinDbg (very convenient to me because I was handling a kernel malware) later. Many attendees asked how I was able to force this dump, so I decided to write a quick explanation for helping who needs this information.

Previously, I had configured my system for creating a manual crash dump when I try the following combination: *right CTRL + BACKSPACE* **(twice the last one)**. Obviously, you can create you own combination, but you need to search a keyboard mapping documentation.

Therefore, the steps for accomplishing are the following:

1. Configure your system for generating either a complete crash dump or a kernel dump. Go to Control Panel -- System (or press **[WINDOWS KEY] + Pause**). There, click on **Advanced System Settings → Startup and Recovery Settings** and configure the system as shown below:



Startup and Recovery

System startup

Default operating system:
Windows 7

☑ Time to display list of operating systems: 30 seconds

☐ Time to display recovery options when needed: 30 seconds

System failure

☑ Write an event to the system log

☑ Automatically restart

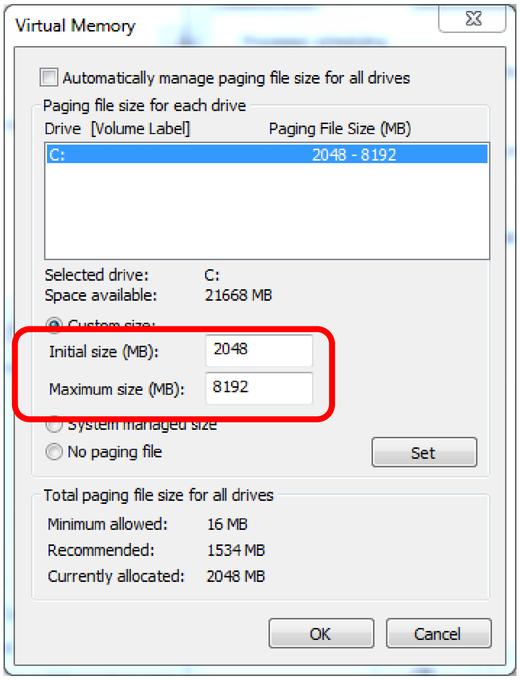
Write debugging information

Complete memory dump

Dump file:
%SystemRoot%\MEMORY.DMP

☑ Overwrite any existing file

2. Verify whether the system has enough space in the page file for the dump. Go to **Control Panel → System** (or press **[WINDOWS KEY] + Pause**). There, go to **Advanced tab → Performance → Settings → Advanced → Change** and make the page file big enough (for example, I usually reserve 50% more than RAM size, but it is not a rule):



Virtual Memory

□ Automatically manage paging file size for all drives

Paging file size for each drive

Drive [Volume Label]	Paging File Size (MB)
C:	2048 - 8192

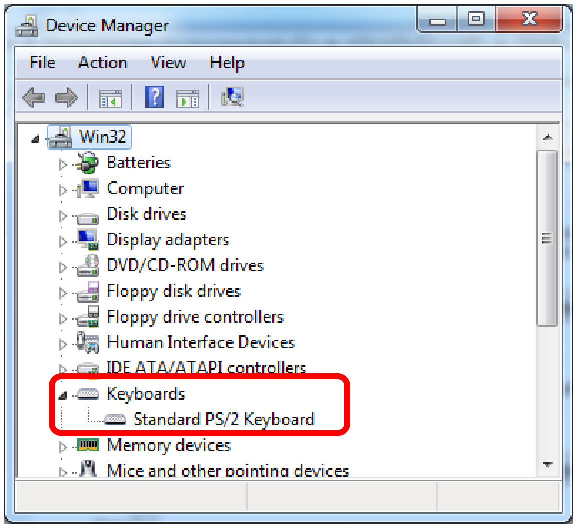
Selected drive: C:
Space available: 21668 MB

● Custom size:
Initial size (MB): 2048
Maximum size (MB): 8192
○ System managed size
○ No paging file

Total paging file size for all drives
Minimum allowed: 16 MB
Recommended: 1534 MB
Currently allocated: 2048 MB

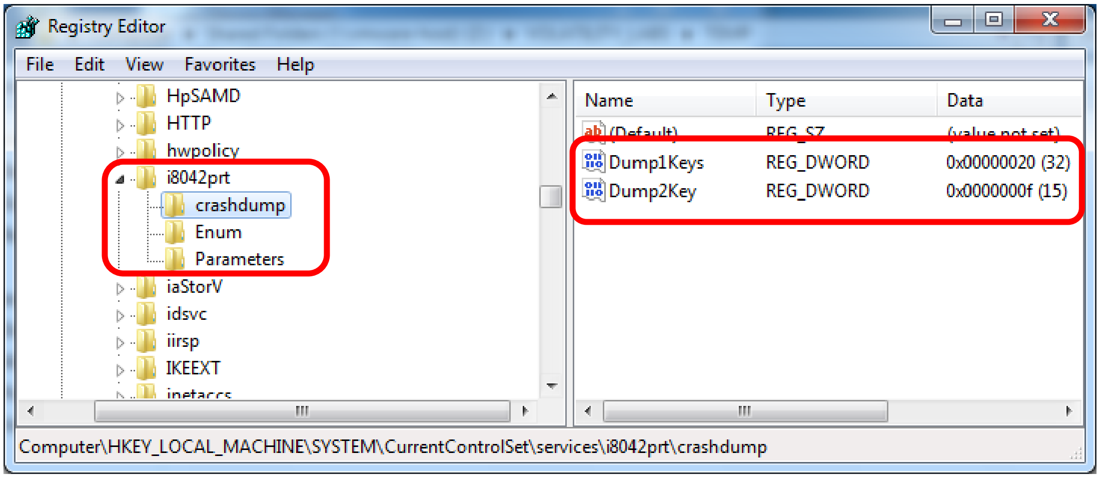
As this system has only 1 GB RAM, so I have reserved 2 GB as minimum and limited the page file at 8 GB. Please, note that I unchecked the "Automatically manage paging file size for all drivers" option.

3. Check whether the keyboard is either PS/2 or USB. Take care here because even modern computers have shown PS/2 keyboard. Thus, go to **Control Panel → System** (or press **[WINDOWS KEY] + Pause**). From there, go to **Hardware tab → Device Manager**. A similar screen should appear:



Win32
- Batteries
- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Floppy disk drives
- Floppy drive controllers
- Human Interface Devices
- IDE ATA/ATAPI controllers
- Keyboards
  - Standard PS/2 Keyboard
- Memory devices
- Mice and other pointing devices

If your system has a **PS/2 keyboard** like mine, so you should create the following keys and values at **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\i8042prt** as shown below:



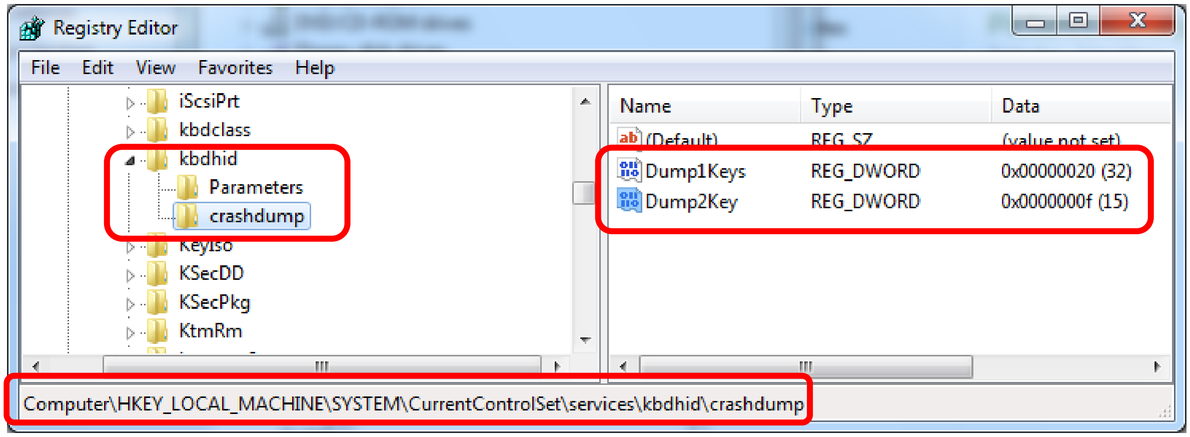
Name	Type	Data
(Default)	REG_SZ	(value not set)
Dump1Keys	REG_DWORD	0x00000020 (32)
Dump2Key	REG_DWORD	0x0000000f (15)

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\i8042prt\crashdump

Where:

- The path is **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\i8042prt**
- The "**i8042prt**" represents PS/2 keyboard.
- The **crashdump** key does not exist by default, so you have to create it.
- The **Dump1Keys** containing the value of **0x20** means the **left CTRL key**.
- The **Dump2Key** containing the value of **0xf** means the **Backspace key** (that must have pressed twice).

If your keyboard is USB, so you have to create the same key and values at **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\kbdhid** as shown below:



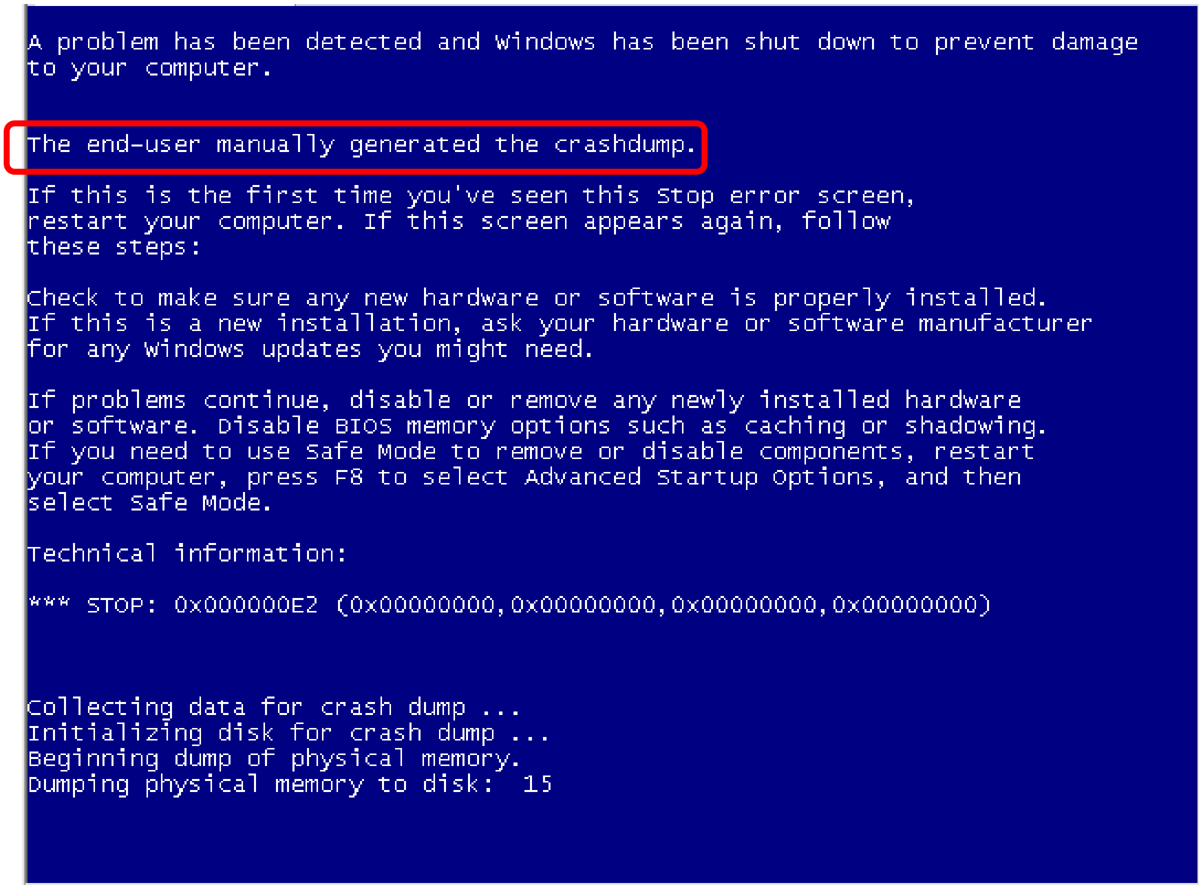
Name	Type	Data
(Default)	REG_SZ	(value not set)
Dump1Keys	REG_DWORD	0x00000020 (32)
Dump2Key	REG_DWORD	0x0000000f (15)

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\kbdhid\crashdump

Where:

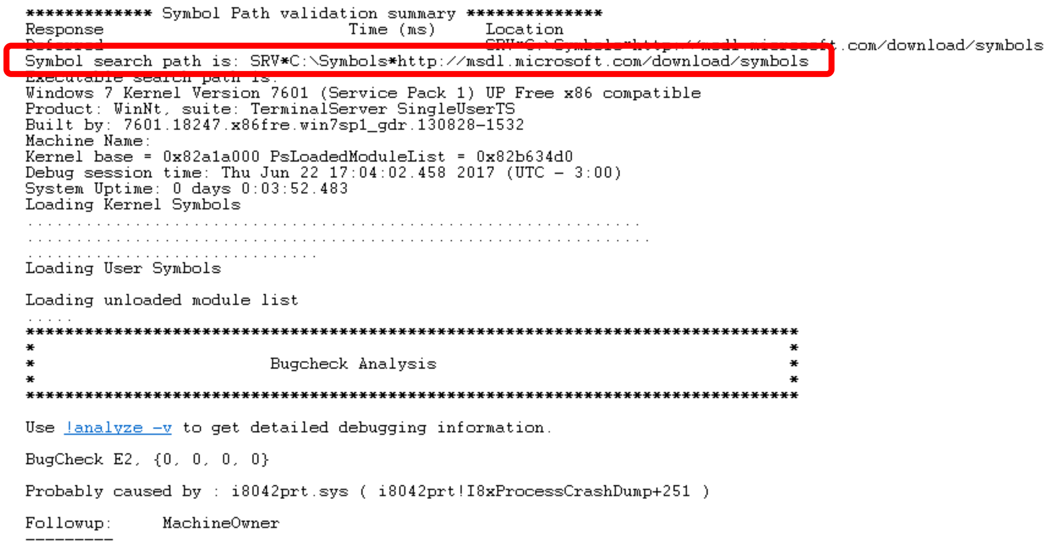
- The path is **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\kbdhid**
- The "**kbdhid**" represents USB keyboard.
- The **crashdump** key does not exist by default, so you have to create it.
- The **Dump1Keys** containing the value of **0x20** means the **left CTRL key**.
- The **Dump2Key** containing the value of **0xf** means the **Backspace key** (that must have pressed **twice**).

4. Finally, to force the configuration to take effect, reboot the system. After the initialization, try to crash the system by pressing **left CTRL + BACKSPACE** (this key must be pressed **twice**). If everything is OK, the system will crash and a file named memory.dmp will be generated at **C:\Windows** directory.

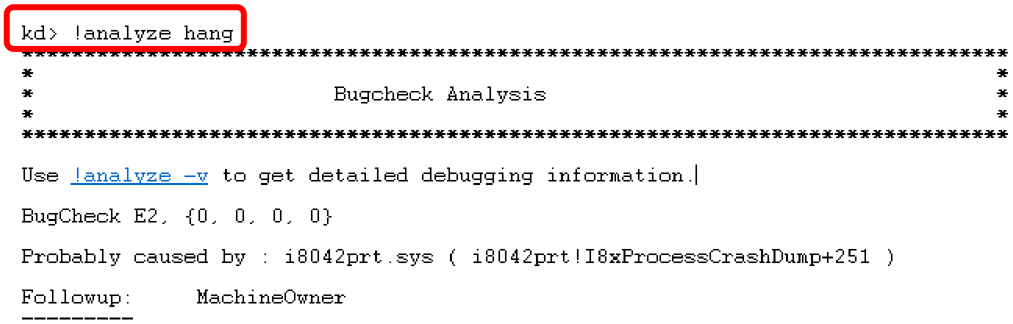


```
A problem has been detected and windows has been shut down to prevent damage
to your computer.

The end-user manually generated the crashdump.

If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow
these steps:

Check to make sure any new hardware or software is properly installed.
If this is a new installation, ask your hardware or software manufacturer
for any Windows updates you might need.

If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use Safe Mode to remove or disable components, restart
your computer, press F8 to select Advanced Startup Options, and then
select Safe Mode.

Technical information:

*** STOP: 0x000000E2 (0x00000000,0x00000000,0x00000000,0x00000000)


Collecting data for crash dump ...
Initializing disk for crash dump ...
Beginning dump of physical memory.
Dumping physical memory to disk:  15
```

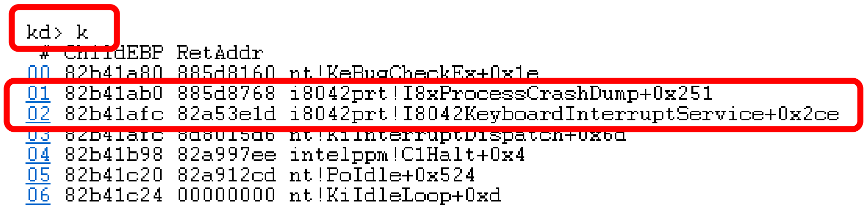
5. We could open the WinDbg and make a last check on this dump. Thus, copy the dump to another folder (**C:\dumps**, for example) for avoiding facing problems with permission. Afterwards, open the **WinDbg** and go to **Open Crash Dump** option. Navigate to **C:\dumps** and choose our dump file (**memory.dmp**) as shown below:



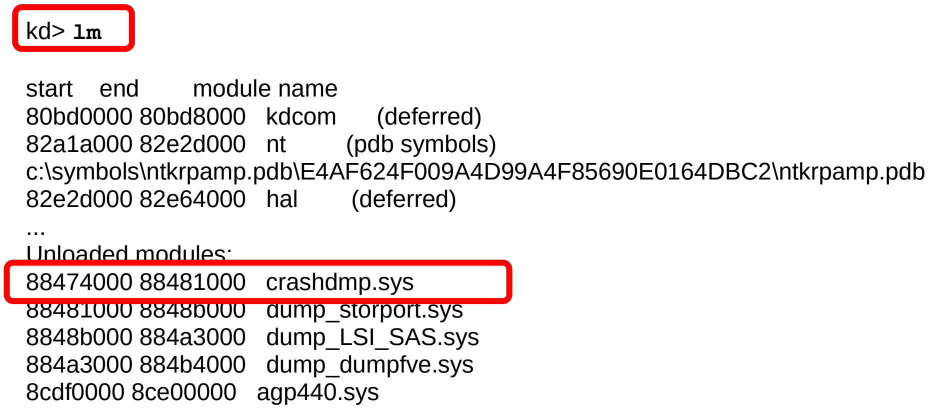
```
************* Symbol Path validation summary **************
Response                         Time (ms)     Location
Deferred                                       SRV*C:\Symbols*http://msdl.microsoft.com/download/symbols
Symbol search path is: SRV*C:\Symbols*http://msdl.microsoft.com/download/symbols
Executable search path is:
Windows 7 Kernel Version 7601 (Service Pack 1) UP Free x86 compatible
Product: WinNt, suite: TerminalServer SingleUserTS
Built by: 7601.18247.x86fre.win7sp1_gdr.130828-1532
Machine Name:
Kernel base = 0x82a1a000 PsLoadedModuleList = 0x82b634d0
Debug session time: Thu Jun 22 17:04:02.458 2017 (UTC - 3:00)
System Uptime: 0 days 0:03:52.483
Loading Kernel Symbols
...............................................................
................................................................
..............................
Loading User Symbols

Loading unloaded module list
.....
*******************************************************************************
*                                                                             *
*                        Bugcheck Analysis                                    *
*                                                                             *
*******************************************************************************

Use !analyze -v to get detailed debugging information.

BugCheck E2, {0, 0, 0, 0}

Probably caused by : i8042prt.sys ( i8042prt!I8xProcessCrashDump+251 )

Followup:     MachineOwner
---------
```



```
kd> !analyze hang
*******************************************************************************
*                                                                             *
*                        Bugcheck Analysis                                    *
*                                                                             *
*******************************************************************************

Use !analyze -v to get detailed debugging information.

BugCheck E2, {0, 0, 0, 0}

Probably caused by : i8042prt.sys ( i8042prt!I8xProcessCrashDump+251 )

Followup:     MachineOwner
---------
```



```
kd> k
 # ChildEBP RetAddr
00 82b41a80 885d8160 nt!KeBugCheckEx+0x1e
01 82b41ab0 885d8768 i8042prt!I8xProcessCrashDump+0x251
02 82b41afc 82a53e1d i8042prt!I8042KeyboardInterruptService+0x2ce
03 82b41afc 8d8015d6 nt!KiInterruptDispatch+0x6d
04 82b41b98 82a997ee intelppm!C1Halt+0x4
05 82b41c20 82a912cd nt!PoIdle+0x524
06 82b41c24 00000000 nt!KiIdleLoop+0xd
```



```
kd> lm

start    end        module name
80bd0000 80bd8000   kdcom      (deferred)
82a1a000 82e2d000   nt         (pdb symbols)
c:\symbols\ntkrpamp.pdb\E4AF624F009A4D99A4F85690E0164DBC2\ntkrpamp.pdb
82e2d000 82e64000   hal        (deferred)
...
Unloaded modules:
88474000 88481000   crashdmp.sys
88481000 8848b000   dump_storport.sys
8848b000 884a3000   dump_LSI_SAS.sys
884a3000 884b4000   dump_dumpfve.sys
8cdf0000 8ce00000   agp440.sys
```

I hope it helps you. Have a nice day.

Alexandre Borges.