

RE: Enumerating IP Addresses

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.development.device.drivers/2006-01/msg00616.html>

- *From:* "Adam" <adamtuliper@xxxxxxxxxxxxxx>
 - *Date:* Mon, 16 Jan 2006 21:21:02 -0800
-

just an fyi, good info at:

<http://www.osronline.com/article.cfm?article=335>

any idea where [esi] points to on your system at that time and what source line it refers to?

--

Adam Tuliper

<http://www.secure-coding.com>

"Gareth" wrote:

```
> I am trying to enumerate all ip addresses from kernel mode, but I get a BSOD
> after registering my callbacks.
>
> I have written a greatly simplified version of the driver, which shows the
> BSOD happening.
>
> Here's the code which is failing:
>
> #include "ntddk.h"
> #include "tdi.h"
> #include "tdikrnl.h"
> #include "tdiinfo.h"
>
>
> #include "stdio.h"
> #include "stdarg.h"
>
> #define RELATIVE(wait) (-(wait))
>
> #define NANOSECONDS(nanos) \
> (((signed __int64)(nanos)) / 100L)
>
> #define MICROSECONDS(micros) \
> (((signed __int64)(micros)) * NANOSECONDS(1000L))
>
```

RE: Enumerating IP Addresses

```
> #define MILLISECONDS(milli) \  
> (((signed __int64)(milli)) * MICROSECONDS(1000L))  
>  
> #define SECONDS(seconds) \  
> (((signed __int64)(seconds)) * MILLISECONDS(1000L))  
>  
> #define MINUTES(minutes) \  
> (((signed __int64)(minutes)) * SECONDS(60L))  
>  
> VOID AddHandler(IN PTA_ADDRESS Address)  
> {  
> DbgPrint("PnP Client AddNetAddress call");  
>  
> if(Address != NULL)  
> {  
> DbgPrint("non empty address found");  
>  
> }  
> }  
>  
> VOID DeleteHandler(IN PTA_ADDRESS Address)  
> {  
> return;  
> }  
>  
>  
>  
> void KSocketEnumerateIPAddresses(void)  
> {  
> NTSTATUS callbackSet;  
> HANDLE bindingHandle = NULL;  
> NTSTATUS didItEnumerate = STATUS_SUCCESS;  
> LARGE_INTEGER waitTime;  
> waitTime.QuadPart= RELATIVE(SECONDS(15));  
>  
> TdiInitialize();  
>  
> // register callback  
> callbackSet = TdiRegisterAddressChangeHandler(AddHandler, DeleteHandler,  
> &bindingHandle);  
>  
> if(callbackSet != STATUS_SUCCESS)  
> {  
> DbgPrint("TdiRegisterPnPHandlers failed");  
> return;  
> }  
>  
> DbgPrint("TdiRegisterPnPHandlers succeeded");  
>  
> // initiate enumeration
```

RE: Enumerating IP Addresses

```
> didItEnumerate = TdiEnumerateAddresses(bindingHandle);
>
> if(callbackSet != STATUS_SUCCESS)
> {
> DbgPrint("TdiEnumerateAddresses failed");
> return;
> }
>
> DbgPrint("TdiEnumerateAddresses succeeded");
>
> KeDelayExecutionThread(KernelMode, FALSE, &waitTime);
> return;
> }
>
> NTSTATUS DriverEntry( IN PDRIVER_OBJECT theDriverObject, IN PUNICODE_STRING
> theRegistryPath )
> {
>
> DbgPrint("DriverEntry called");
>
> KSocketEnumerateIPAddresses();
>
> return STATUS_SUCCESS;
> }
>
> The crash output from windbg is as follows:
>
> DriverEntry called
> PnP Client AddNetAddress call
> non empty address found
> PnP Client AddNetAddress call
> non empty address found
> TdiRegisterPnPHandlers succeeded
> PnP Client AddNetAddress call
> non empty address found
>
> *** Fatal System Error: 0x00000050
> (0xFFFF833E8,0x00000000,0xFFFF833E8,0x00000000)
>
> Break instruction exception – code 80000003 (first chance)
>
> A fatal system error has occurred.
> Debugger entered on first try; Bugcheck callbacks have not been invoked.
>
> A fatal system error has occurred.
>
> Connected to Windows XP 2600 x86 compatible target, ptr64 FALSE
> Loading Kernel Symbols
> .....
> Loading unloaded module list
> .....
```

RE: Enumerating IP Addresses

```
> Loading User Symbols
> *****
> *
> *
> * Bugcheck Analysis
> *
> *
> *
> *****
>
> Use !analyze -v to get detailed debugging information.
>
> BugCheck 50, {fff833e8, 0, fff833e8, 0}
> kd> !analyze -v
> *****
> *
> *
> * Bugcheck Analysis
> *
> *
> *
> *****
>
> PAGE_FAULT_IN_NONPAGED_AREA (50)
> Invalid system memory was referenced. This cannot be protected by
> try-except,
> it must be protected by a Probe. Typically the address is just plain bad or
> it
> is pointing at freed memory.
> Arguments:
> Arg1: fff833e8, memory referenced.
> Arg2: 00000000, value 0 = read operation, 1 = write operation.
> Arg3: fff833e8, If non-zero, the instruction address which referenced the
> bad memory
> address.
> Arg4: 00000000, (reserved)
>
> Debugging Details:
> -----
>
>
> READ_ADDRESS: fff833e8
>
> FAULTING_IP:
> +ffffffffff833e8
> fff833e8 ?? ???
>
> MM_INTERNAL_CODE: 0
>
> DEFAULT_BUCKET_ID: INTEL_CPU_MICROCODE_ZERO
>
```

RE: Enumerating IP Addresses

```
> BUGCHECK_STR: 0x50
>
> LAST_CONTROL_TRANSFER: from fc95c0a8 to fff833e8
>
> TRAP_FRAME: fc906b70 -- (.trap ffffffff906b70)
> ErrCode = 00000000
> eax=fff833e8 ebx=fc95dc30 ecx=80dd5120 edx=00000017 esi=80dd50e0
> edi=fc95dc28
> eip=fff833e8 esp=fc906be4 ebp=00000000 iopl=0         nv up ei ng nz na po
> nc
> cs=0008  ss=0010  ds=0023  es=0023  fs=0030  gs=0000
> efl=00010286
> fff833e8 ?? ???
> Resetting default scope
>
> STACK_TEXT:
> WARNING: Frame IP not in any known module. Following frames may be wrong.
> fc906be0 fc95c0a8 80dd5120 80dd5100 ffb4abe0 0xffff833e8
> fc906c00 fc95ce11 fc95dc28 80cf5308 ffb9f038 TDI!TdiNotifyAddresses+0x28
> fc906c20 fc95d05e 00000000 ffb9f000 80d9f328 TDI!TdiExecuteRequest+0x353
> fc906c5c fc95d67f 80cf5308 00000010 fa506578
> TDI!TdiHandleSerializedRequest+0x1bc
> fc906c68 fa506578 80cf5308 00000000 f70f2e80 TDI!TdiEnumerateAddresses+0xb
> fc906c88 fa5065e5 fc906ce4 805acfe9 80d9f328
> myDriver!KSocketEnumerateIPAddresses+0x68 [z
> \winddk\enumerateip\enumerate.c @ 68]
> fc906c90 805acfe9 80d9f328 ff5c8000 00000000 myDriver!DriverEntry+0x15 [z
> \winddk\enumerateip\enumerate.c @ 89]
> fc906d4c 805a7475 00000910 ff5c8000 80d9f328 nt!IopLoadDriver+0x5e0
> fc906d74 804ebd08 00000910 00000000 80e908b8 nt!IopLoadUnloadDriver+0x43
> fc906dac 80559026 fa876cf4 00000000 00000000 nt!ExpWorkerThread+0xfe
> fc906ddc 8050f513 804ebc35 00000001 00000000 nt!PspSystemThreadStartup+0x34
> 00000000 00000000 00000000 00000000 00000000 nt!KiThreadStartup+0x16
>
>
> FAILED_INSTRUCTION_ADDRESS:
> +fffffffffff833e8
> fff833e8 ?? ???
>
> FOLLOWUP_IP:
> TDI!TdiNotifyAddresses+28
> fc95c0a8 8b36 mov esi,[esi]
>
> SYMBOL_STACK_INDEX: 1
>
> FOLLOWUP_NAME: MachineOwner
>
> SYMBOL_NAME: TDI!TdiNotifyAddresses+28
>
> MODULE_NAME: TDI
>
```

RE: Enumerating IP Addresses

> IMAGE_NAME: TDI.SYS
>
> DEBUG_FLR_IMAGE_TIMESTAMP: 3b7d8535
>
> STACK_COMMAND: .trap ffffffff906b70 ; kb
>
> FAILURE_BUCKET_ID: 0x50_CODE_AV_BAD_IP_TDI!TdiNotifyAddresses+28
>
> BUCKET_ID: 0x50_CODE_AV_BAD_IP_TDI!TdiNotifyAddresses+28
>
> Followup: MachineOwner
> -----
>
> Can anyone suggest what I am doing wrong? It seems to work, as the callback
> is returning ip addresses, but something is causing TDI!TdiNotifyAddresses
> to reference an invalid pointer...
>
> Many thanks in advance.
>
>
>
>

• *Follow-Ups:*

- ◆ [RE: Enumerating IP Addresses](#)
 ◇ From: Gareth

• *References:*

- ◆ [Enumerating IP Addresses](#)
 ◇ From: Gareth
- Prev by Date: [Re: Unloading driver after DriverEntry fails](#)
- Next by Date: [RE: Enumerating IP Addresses](#)
- Previous by thread: [Enumerating IP Addresses](#)
- Next by thread: [RE: Enumerating IP Addresses](#)
- Index(es):
 - ◆ [Date](#)
 - ◆ [Thread](#)