

Re: How to get Parent Pid from Kernel Driver?

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.kernel/2006-04/msg00527.htm>

- *From:* macro <macro913@xxxxxxxx>
 - *Date:* Thu, 20 Apr 2006 08:32:50 +0800
-

Norman Diamond ^{TMS}:

In the documented portions of NtQueryInformationProcess and NtQuerySystemInformation (which Microsoft says they might change all the time in the future), I do not see any way to get the Parent Pid. For a user mode process to get this kind of information from the kernel, except for NT4, I called:

GetWindowThreadProcessId
CreateToolhelp32Snapshot
Process32First
Process32Next (in a loop)
CloseHandle

For a driver to get this kind of information, I don't know the answer.

"anton bassov" <xxx@xxxxxxx> wrote in message
news:a914728685fa4820a94fcfd28c4c1323@xxxxxxxxxxxxxxxx

Hi mate

I often advise "unsupported" things, but here is the exception to the rule – I would not advise you to access such structures as EPROCESS,ETHREAD,etc (Microsoft changes them all the time). Therefore, I would rather advise you to call NtQueryInformationProcess() or NtQuerySystemInformation()

Regards

Anton Bassov

Through the EPROCESS and the PEB struct,i finished the task to get the parent pid and the full path.But i used a lot of hardcode in the struct.So i wanted to get a more universal method ,Can anyone can help me? 3ks.This is my code.

```
typedef struct ot_entry {  
    ULONG signature;  
    struct ot_entry *next;  
    PDEVICE_OBJECT devobj;  
    PFILE_OBJECT fileobj, associated_fileobj;
```

Re: How to get Parent Pid from Kernel Driver?

```
int type;
ULONG out_offset, in_offset, out_oob_offset, in_oob_offset;
int flt_rule;
ULONG pid;
ULONG ppid;
char PProcessName[16];
//char ProcessPath[512];
char ProcessPath[256];
char ProcessName[16];
UCHAR ippoto; // Protocol for this connection
} ot_entry_t;
void GetProcessInfo2k(ot_entry_t *one)
{
PEPROCESS curproc;
NTSTATUS status;

PCHAR work, help, go;
ULONG pid, now;
USHORT num;
//ANSI_STRING name;
int i;

PCHAR nameptr;
one->pid = (ULONG)PsGetCurrentProcessId();
curproc = PsGetCurrentProcess();
work = (PCHAR)curproc;

one->ppid = *((ULONG *)(work + 0x1c8));
DbgPrint("The ppid is %d\n", one->ppid);
nameptr = (PCHAR)curproc + 0x1fc;
strncpy(one->ProcessName, nameptr, 16);

DbgPrint("The ProcessName is %s\n", one->ProcessName);
DbgPrint("The ProcessID is %d\n", one->pid);
if(one->ppid != one->pid)
{
work = (PCHAR)curproc + 0xa0;
work = (PCHAR)(((LIST_ENTRY *)work)->Flink);
work = work - 0xa0;
while(work != (PCHAR)curproc)
{
help = work;
pid = *((ULONG *)(help + 0x9c));
DbgPrint("Pid is %d\n", pid);
if(pid == one->ppid)
{
nameptr = help + 0x1fc;
strncpy(one->PProcessName, nameptr, 16);
DbgPrint("The Parent ProcessName is %s\n", one->PProcessName);
break;
}
}
```

Re: How to get Parent Pid from Kernel Driver?

Re: How to get Parent Pid from Kernel Driver?

```
work = work + 0xa0;
work = (PCHAR)((LIST_ENTRY *)work)->Flink;
work = work - 0xa0;
}
}
else
{
strncpy(one->PProcessName, one->ProcessName, 16);
}
if((one->pid == 0) || (one->pid == 8))
{
DbgPrint("It is system!\n");
return;
}
//
else //if((one->pid != 0) && (one->pid != 8))
{
work = (PCHAR)curproc + 0x1b0;
now = *((ULONG *)work);
now = now + 0x10;
now = *((ULONG *)now); //segment address
now = now + 0x38; //PFILE_OBJECT
DbgPrint("05 now is :%x\n", now);
//DbgPrint(("The full path is:%ws\n", ((UNICODE_STRING *)now)->Buffer));
/*RtlUnicodeStringToAnsiString(&name, (UNICODE_STRING *)now, TRUE);
DbgPrint(("O6\n"));
strncpy(one->ProcessPath, (PCHAR)(name.Buffer), 256);
DbgPrint(("the full path is:%s\n", one->ProcessPath));
DbgPrint(("OK!\n"));
ExFreePool(name.Buffer);*/
num = *((USHORT *)now);
DbgPrint("num is %d\n", num);
now = now + 4;
now = *((ULONG *)now);
go = (PCHAR)now;
for(i = 0; i < num; i++)
{
one->ProcessPath[i] = *go;
go = go + 2;
}
one->ProcessPath[i] = 0;
go = one->ProcessPath;
DbgPrint("the process full path is: %s\n", one->ProcessPath);
}
}
.
```