

KeServiceDescriptorTable

Peter Kleissner

June 09, 2009

This document explains the internal structure `KeServiceDescriptorTable` in the Windows Kernel and its subsidiary tables. It is responsible for handling (dispatching) internal Windows calls (using `sysenter/syscall` and `int 2Eh`).

There are interesting `ntoskrnl.exe` exports:

```
710  2AC 0014A430 KeAddSystemServiceTable
816  31D 00275E81 KeRemoveSystemServiceTable
824  325 0012C8C0 KeServiceDescriptorTable
```

`KeServiceDescriptorTable` is the data structure (an array) that holds service tables. A service table is used to lookup a function number and for dispatching the function number to a function. Windows knows currently only two: one of `ntoskrnl.exe` and one of `win32k.sys`. It has been reported that there are 4 slots preserved (and always only 2 of them used) under some Windows versions (changes with version and service pack).

A service table is added by `KeAddSystemServiceTable` and deleted by `KeRemoveSystemServiceTable` (however I have never seen the later one used). So the format of the `KeServiceDescriptorTable` (and its entries) is following:

```
struct KeServiceDescriptorTable
{
    System_Service_Table ntoskrnl;    // ntoskrnl.exe (native API)
    System_Service_Table win32k;      // win32k.sys (gdi/user support)
    System_Service_Table Table3;      // (unused)
    System_Service_Table Table4;      // (unused)
};

struct System_Service_Table
{
    PNTPROC ServiceTable;    // array of entry points
    PDWORD CounterTable;    // array of usage counters (= 0)
    DWORD ServiceLimit;     // number of table entries
    PBYTE ArgumentTable;    // array of byte counts
};
```

Let's take a look at a memory dump of the Service Descriptor Table (SDT):

```
KeServiceDescriptorTable:
808aeee0 8083fc4c ntoskrnl!KiServiceTable
808aeee4 00000000
808aeee8 00000128
808aeeec 80803618 ntoskrnl!KiArgumentTable
808aeeef0 a01859f0 win32k!W32pServiceTable
```

```
808aeef4 00000000
808aeef8 0000027f
808aeefc a0186670 win32k!W32pArgumentTable
```

The dump shows the first and second entry of the SDT:

```
KiServiceTable = System Service Dispatch Table (SSDT)
KiArgumentTable = System Service Parameter Table (SSPT)
```

The SSDT is very often used for malware development to hook Windows functions.

It is important to say that there is a `KeServiceDescriptorTableShadow` which is not exported. It contains both `ntoskrnl` and `win32k` entries while the main table only maintains the `ntoskrnl` one. Every thread gets the `KeServiceDescriptorTable` pointer into his `Thread Control Block`, however it is possible to create an own SDT for any thread.

So how is the Service Dispatch Table used by Windows?

- `sysenter / syscall`
- `int 2Eh`
- direct call to `KiSystemService()`

`KiSystemService()` (also not exported) handles a service call (independent from raised by `sysenter/syscall`, `int 2Eh` or direct call) and dispatches the function number passed. The Windows 2000 Native API document [3] explains very nice at page 10 "THE INT 2Eh SYSTEM SERVICE HANDLER" what the function exactly does.

Finally, Stoned also adds its own System Service Table for installing live the new Stoned Subsystem to Windows.

References:

- [1] RE: How to dump system service dispatch table?
<http://www.tech-archive.net/pdf/Archive/Development/microsoft.public.win32.programmer.kernel/2008-04/msg00290.pdf>
- [2] Notes from "Windows NT System-Call Hooking" (Dr. Dobb's Journal, '97)
[nt_hooking.txt](http://www.dobbsjournal.com/articles/1997/09/nt_hooking.txt)
- [3] The Windows 2000 Native API
<http://undocumented.rawol.com/sbs-w2k-2-the-windows-2000-native-api.pdf>