
# Windows API Exploitation Recipes for Red – Blue Teams

Vivek Ramachandran

Chief Trainer, Pentester Academy

http://PentesterAcademy.com

# Process Listing APIs: CreateToolhelp32Snapshot

# Tool Help Library: CreateToolhelp32Snapshot

## CreateToolhelp32Snapshot function

Takes a snapshot of the specified processes, as well as the heaps, modules, and threads used by these processes.

### Syntax

C++

```
HANDLE WINAPI CreateToolhelp32Snapshot(
  _In_ DWORD dwFlags,
  _In_ DWORD th32ProcessID
);
```

https://msdn.microsoft.com/en-us/library/windows/desktop/ms682489(v=vs.85).aspx

# TH32CS_SNAPPROCESS

| | |
|---|---|
| **TH32CS_SNAPPROCESS** 0x00000002 | Includes all processes in the system in the snapshot. To enumerate the processes, see **Process32First**. |

# Let’s get some process data!

## PROCESSENTRY32 structure

Describes an entry from a list of the processes residing in the system address space when a snapshot was taken.

### Syntax

C++

```
typedef struct tagPROCESSENTRY32 {
  DWORD     dwSize;
  DWORD     cntUsage;
  DWORD     th32ProcessID;
  ULONG_PTR th32DefaultHeapID;
  DWORD     th32ModuleID;
  DWORD     cntThreads;
  DWORD     th32ParentProcessID;
  LONG      pcPriClassBase;
  DWORD     dwFlags;
  TCHAR     szExeFile[MAX_PATH];
} PROCESSENTRY32, *PPROCESSENTRY32;
```

https://msdn.microsoft.com/en-us/library/windows/desktop/ms684839(v=vs.85).aspx

# Note: Can do a LOT More!

| Value | Meaning |
|---|---|
| **TH32CS_INHERIT** 0x80000000 | Indicates that the snapshot handle is to be inheritable. |
| **TH32CS_SNAPALL** | Includes all processes and threads in the system, plus the heaps and modules of the process specified in *th32ProcessID*. Equivalent to specifying the **TH32CS_SNAPHEAPLIST**, **TH32CS_SNAPMODULE**, **TH32CS_SNAPPROCESS**, and **TH32CS_SNAPTHREAD** values combined using an OR operation ('\|'). |
| **TH32CS_SNAPHEAPLIST** 0x00000001 | Includes all heaps of the process specified in *th32ProcessID* in the snapshot. To enumerate the heaps, see **Heap32ListFirst**. |
| **TH32CS_SNAPMODULE** 0x00000008 | Includes all modules of the process specified in *th32ProcessID* in the snapshot. To enumerate the modules, see **Module32First**. If the function fails with **ERROR_BAD_LENGTH**, retry the function until it succeeds.<br><br>**64-bit Windows:** Using this flag in a 32-bit process includes the 32-bit modules of the process specified in *th32ProcessID*, while using it in a 64-bit process includes the 64-bit modules. To include the 32-bit modules of the process specified in *th32ProcessID* from a 64-bit process, use the **TH32CS_SNAPMODULE32** flag. |
| **TH32CS_SNAPMODULE32** 0x00000010 | Includes all 32-bit modules of the process specified in *th32ProcessID* in the snapshot when called from a 64-bit process. This flag can be combined with **TH32CS_SNAPMODULE** or **TH32CS_SNAPALL**. If the function fails with **ERROR_BAD_LENGTH**, retry the function until it succeeds. |
| **TH32CS_SNAPPROCESS** 0x00000002 | Includes all processes in the system in the snapshot. To enumerate the processes, see **Process32First**. |
| **TH32CS_SNAPTHREAD** 0x00000004 | Includes all threads in the system in the snapshot. To enumerate the threads, see **Thread32First**.<br><br>To identify the threads that belong to a specific process, compare its process identifier to the **th32OwnerProcessID** member of the **THREADENTRY32** structure when enumerating the threads. |

# Do we need Privileges?

- For PID, EXE name etc. basic information we do not need elevation or SeDebugPrivilege ☺
- Keeps our call under the radar ☺
  - Check what we can find for all process enumeration APIs we discuss without requiring elevation or SeDebugPrivilege

# Who else uses it? ☺

- Zeus
  http://www.ioactive.com/pdfs/ZeusSpyEyeBankingTrojanAnalysis.pdf
- PoS Malware
  http://www.trendmicro.co.nz/cloud-content/nz/pdfs/security-intelligence/white-papers/wp-pos-ram-scraper-malware.pdf
- … many many more!
- Used by a lot of system monitoring and performance gathering utilities
- Not considered malicious by AVs

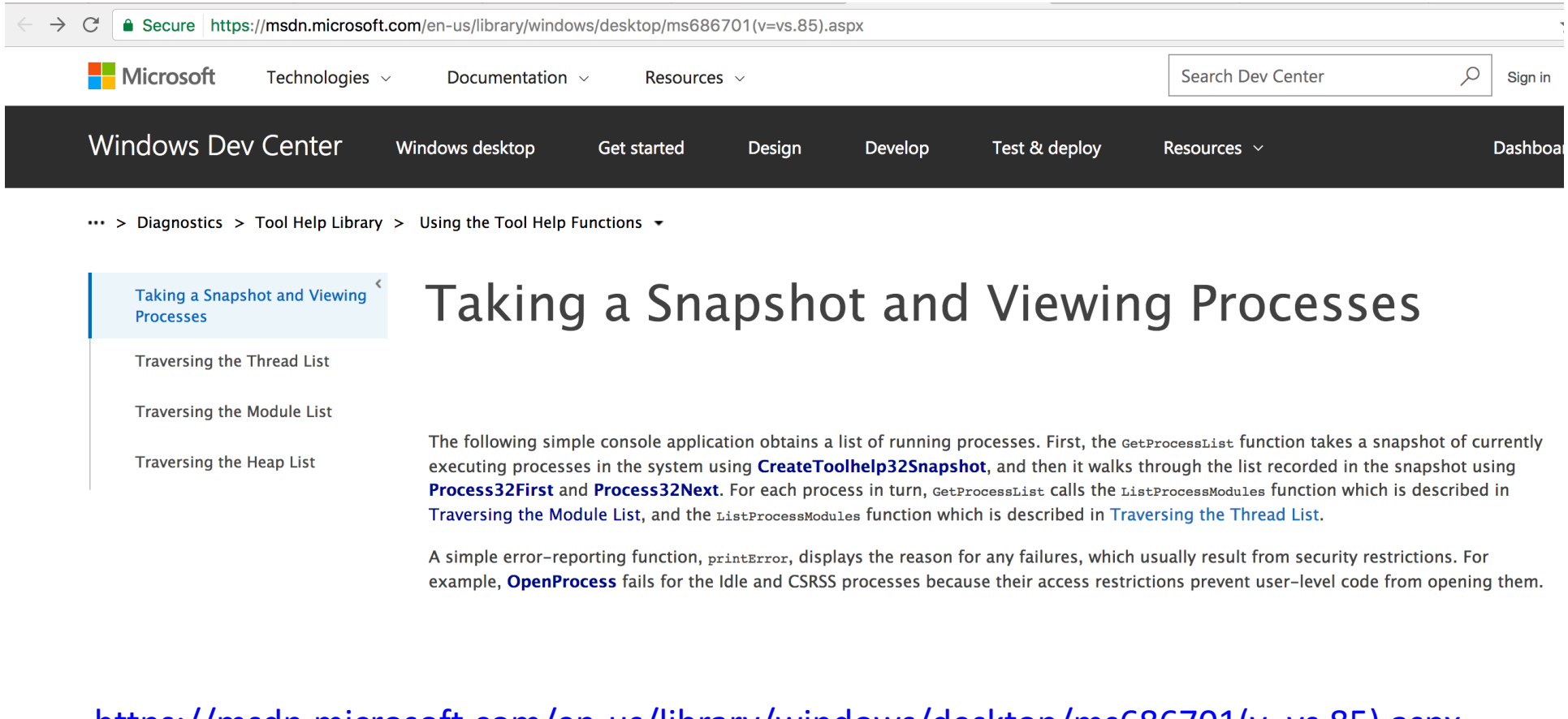
# Advantages

- Quickly figure out what is running
- AV, HIDS etc. security solutions have predictable EXE names
- Can decide if:
  - Run further
  - Which evasion to use?
  - Or maybe even exit?

# Downsides?

- It creates a snapshot!
- While you are analyzing the data:
  - New processes might be created
  - Older processes might be destroyed
  - Parent-Child relationships might change
  - …
- From a Red Teaming perspective this is generally ok
- Need Kernel Mode to have “live” visibility

# Exercises!




https://msdn.microsoft.com/en-us/library/windows/desktop/ms686701(v=vs.85).aspx

# Thank You!