Intro to the EPROCESS

Experiment: Hiding a process from Task Manager by manually patching the Windows Kernel using WinDBG

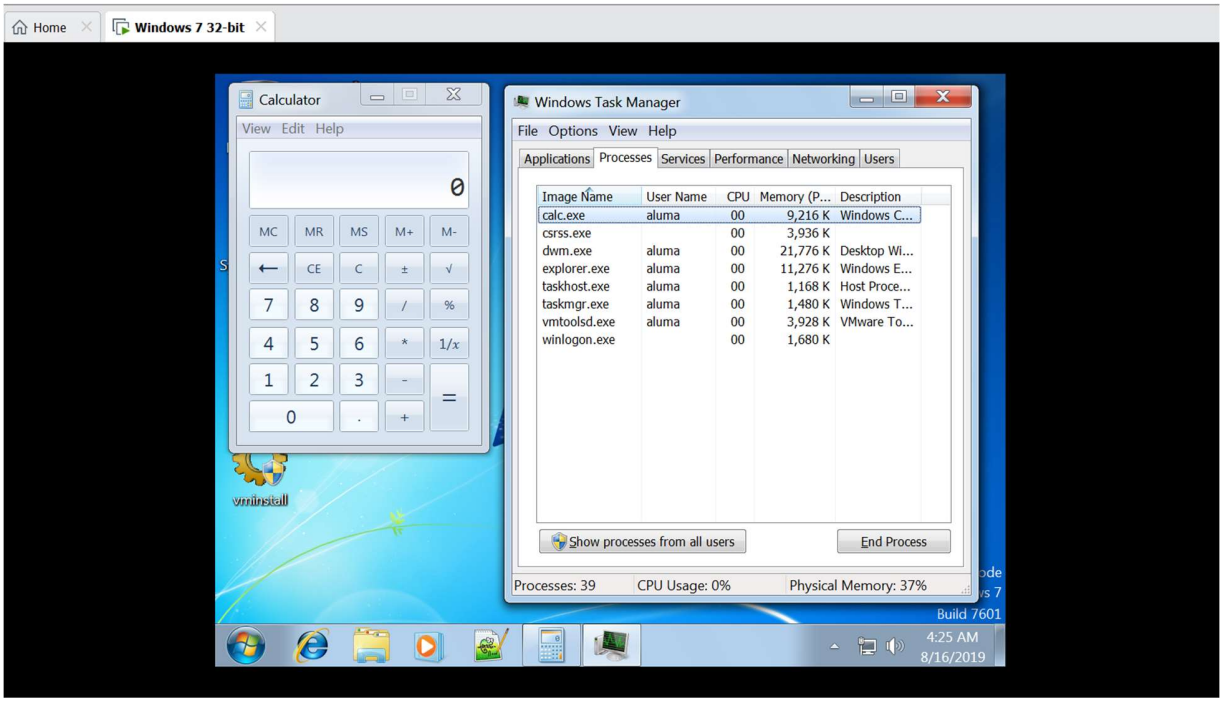
Background

- One of the structures that represents a process in the memory is the **EPROCESS**, which contains information needed to the system for using and handling the process, such as threads information, flags, file name, base address, security tokens, etc.
- One of the fields inside this structure is the **ActiveProcessLink** field, which is used to link the current EPROCESS to the other EPROCESSs, as part of a doubly-linked list. The ActiveProcessLink field is a **LIST_ENTRY**[1] structure that contains both links: forwards and backwards.
- Task Manager (taskmgr.exe) uses WINAPI functions that relies on this linked list structure in order to update its **Processes** tab.
- We will change the relevant fields in the memory in order to make taskmgr.exe to hide calc.exe from the Processes window, even though calc.exe is running.
- We will use WinDBG[2] to remotely debug a Virtual machine with Windows 7 32-bit OS.

Let's go!

We start by opening the virtual machine, then open calc.exe and taskmgr.exe with its Processes tab:

[1] https://docs.microsoft.com/en-us/windows/win32/api/ntdef/ns-ntdef-list_entry
[2] https://docs.microsoft.com/en-us/windows-hardware/drivers/debugger/debugger-download-tools



So far – as expected.

We return control to the debugger and run the command "**!process 0 0**" to get a list of the process running in the machine:

```
kd> !process 0 0
**** NT ACTIVE PROCESS DUMP ****
PROCESS 843d39e8  SessionId: none  Cid: 0004    Peb: 00000000  ParentCid: 0000
    DirBase: 00185000  ObjectTable: 87e01b88  HandleCount: 470.
    Image: System

PROCESS 84f42d40  SessionId: none  Cid: 00f4    Peb: 7ffdf000  ParentCid: 0004
    DirBase: 3f343020  ObjectTable: 8a6cde98  HandleCount:  29.
    Image: smss.exe

PROCESS 84efb030  SessionId: 0  Cid: 0148    Peb: 7ffda000  ParentCid: 013c
    DirBase: 3f343060  ObjectTable: 8dd19570  HandleCount: 444.
    Image: csrss.exe

PROCESS 85856030  SessionId: 1  Cid: 017c    Peb: 7ffde000  ParentCid: 0174
    DirBase: 3f3430a0  ObjectTable: 8ed269c0  HandleCount: 176.
    Image: csrss.exe

PROCESS 85859d40  SessionId: 0  Cid: 019c    Peb: 7ffd4000  ParentCid: 013c
    DirBase: 3f3430c0  ObjectTable: 8ed42c90  HandleCount:  75.
    Image: wininit.exe

PROCESS 85860b90  SessionId: 1  Cid: 01a4    Peb: 7ffd5000  ParentCid: 0174
    DirBase: 3f343040  ObjectTable: 87e64628  HandleCount: 112.
    Image: winlogon.exe

PROCESS 85b94030  SessionId: 0  Cid: 01ec    Peb: 7ffd6000  ParentCid: 019c
    DirBase: 3f343080  ObjectTable: 88611e68  HandleCount: 201.
    Image: services.exe

PROCESS 85b9f030  SessionId: 0  Cid: 01fc    Peb: 7ffd8000  ParentCid: 019c
    DirBase: 3f3430e0  ObjectTable: 9665af08  HandleCount: 526.
    Image: lsass.exe

PROCESS 85ba0478  SessionId: 0  Cid: 0204    Peb: 7ffd7000  ParentCid: 019c
    DirBase: 3f343100  ObjectTable: 88629b58  HandleCount: 137.
    Image: lsm.exe

PROCESS 85bc8c08  SessionId: 0  Cid: 0270    Peb: 7ffdf000  ParentCid: 01ec
```

Locating the process that we are interested in:

Intro to the EPROCESS by Aluma010: https://github.com/Aluma010/Intro-to-the-EPROCESS

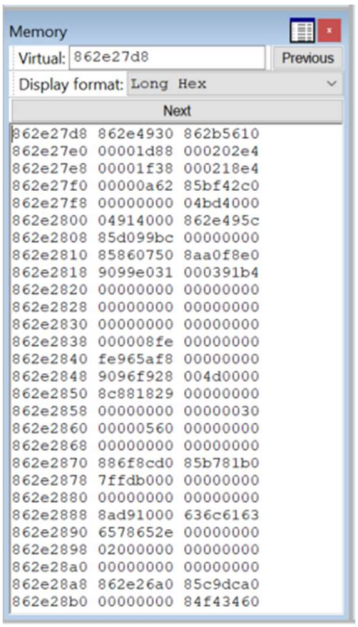
```
PROCESS 862afd40  SessionId: 0  Cid: 08b8    Peb: 7ffd7000  ParentCid: 086c
    DirBase: 3f343420  ObjectTable: 95234c18  HandleCount: 235.
    Image: SearchProtocolHost.exe

PROCESS 862b5558  SessionId: 0  Cid: 08cc    Peb: 7ffdf000  ParentCid: 086c
    DirBase: 3f343440  ObjectTable: 952401a0  HandleCount:  75.
    Image: SearchFilterHost.exe

PROCESS 862e2720  SessionId: 1  Cid: 0944    Peb: 7ffdc000  ParentCid: 0560
    DirBase: 3f343460  ObjectTable: 8aa0f8e0  HandleCount:  74.
    Image: calc.exe

PROCESS 862e4878  SessionId: 1  Cid: 0960    Peb: 7ffd8000  ParentCid: 01a4
    DirBase: 3f343480  ObjectTable: 94af83c0  HandleCount: 108.
    Image: taskmgr.exe

PROCESS 84e8ab40  SessionId: 0  Cid: 0988    Peb: 7ffdf000  ParentCid: 0270
    DirBase: 3f3434a0  ObjectTable: 909db070  HandleCount: 294.
    Image: WmiPrvSE.exe
```

We can see the calc.exe process, with its EPROCESS located in address 0x862e2720.

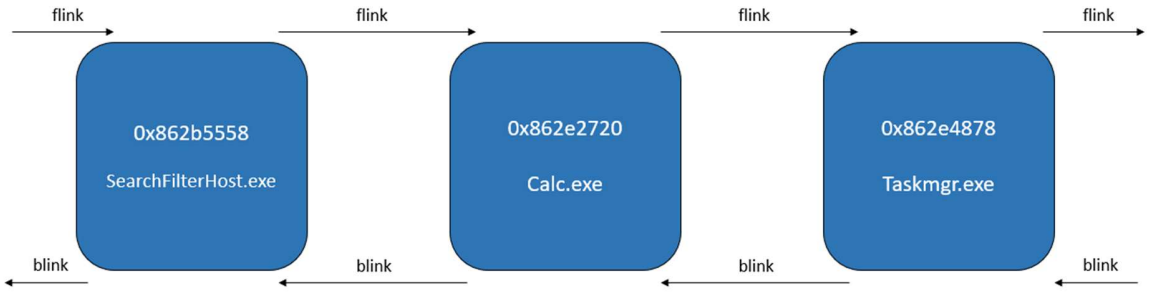
Let's examine the structure of the EPROCESS by running "**dt nt!_eprocess**":

```
kd> dt nt!_eprocess
   +0x000 Pcb              : _KPROCESS
   +0x098 ProcessLock      : _EX_PUSH_LOCK
   +0x0a0 CreateTime       : _LARGE_INTEGER
   +0x0a8 ExitTime         : _LARGE_INTEGER
   +0x0b0 RundownProtect   : _EX_RUNDOWN_REF
   +0x0b4 UniqueProcessId  : Ptr32 Void
   +0x0b8 ActiveProcessLinks : _LIST_ENTRY
   +0x0c0 ProcessQuotaUsage : [2] Uint4B
   +0x0c8 ProcessQuotaPeak : [2] Uint4B
   +0x0d0 CommitCharge     : Uint4B
   +0x0d4 QuotaBlock       : Ptr32 _EPROCESS_QUOTA_BLOCK
   +0x0d8 CpuQuotaBlock    : Ptr32 _PS_CPU_QUOTA_BLOCK
   +0x0dc PeakVirtualSize  : Uint4B
   +0x0e0 VirtualSize      : Uint4B
   +0x0e4 SessionProcessLinks : _LIST_ENTRY
   +0x0ec DebugPort        : Ptr32 Void
   +0x0f0 ExceptionPortData : Ptr32 Void
   +0x0f0 ExceptionPortValue : Uint4B
   +0x0f0 ExceptionPortState : Pos 0, 3 Bits
   +0x0f4 ObjectTable      : Ptr32 _HANDLE_TABLE
   +0x0f8 Token            : _EX_FAST_REF
   +0x0fc WorkingSetPage   : Uint4B
   +0x100 AddressCreationLock : _EX_PUSH_LOCK
   +0x104 RotateInProgress : Ptr32 _ETHREAD
   +0x108 ForkInProgress   : Ptr32 _ETHREAD
   +0x10c HardwareTrigger  : Uint4B
   +0x110 PhysicalVadRoot  : Ptr32 _MM_AVL_TABLE
   +0x114 CloneRoot        : Ptr32 Void
   +0x118 NumberOfPrivatePages : Uint4B
   +0x11c NumberOfLockedPages : Uint4B
```

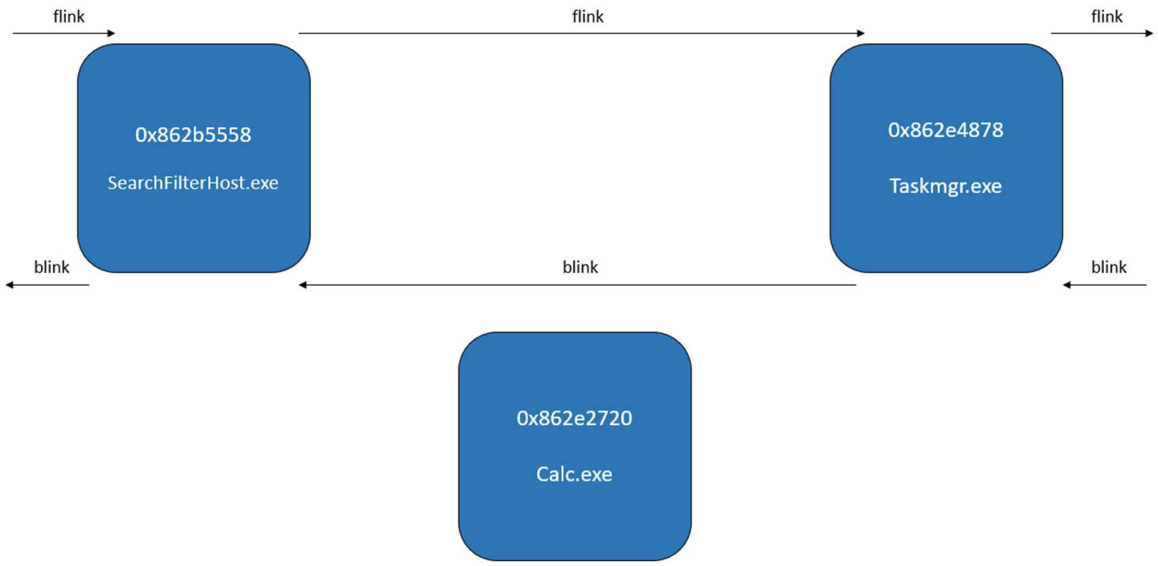
We see that the ActiveProcessLink(s) field is located 0xb8 bytes after the beginning of struct, therefore the ActiveProcessLink of calc.exe is located in 0x862e2720 + 0xb8 = 0x862e27d8. Let's look at it using a memory window of WinDBG:



So: the flink of calc.exe is 0x862e4930, which is Taskmgr.exe (because 0x862e4930 – 0xb8 = 0x862e4878), and the blink of calc.exe is 0x862b5610, which is SearchFilterHost.exe (because 0x862b5610 – 0xb8 = 0x862b5558). Like this:

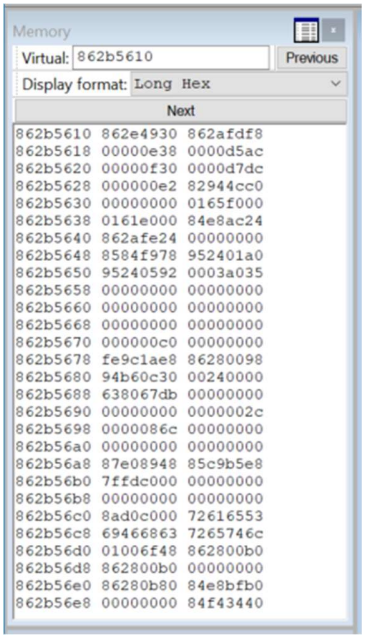


We want to change the list so calc.exe will be left out, like this:



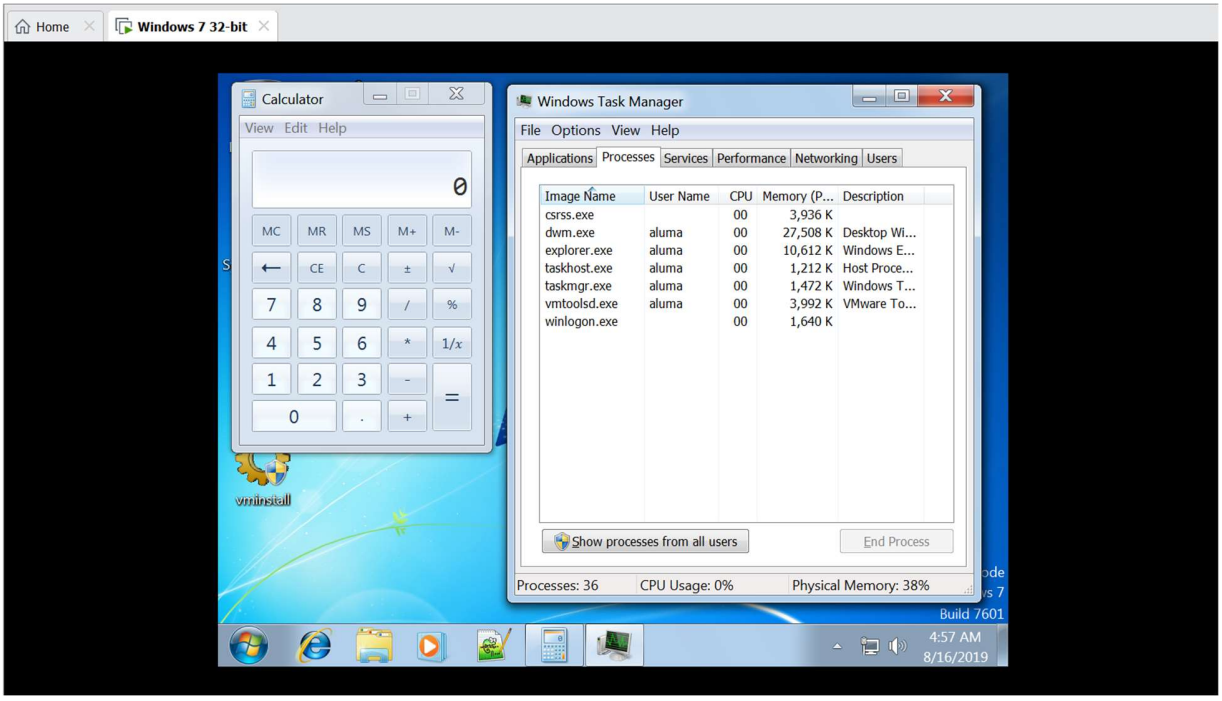
We need to change two fields: the flink of SearchFilterHost.exe, and the blink of Taskmgr.exe.

We open the memory window in WinDBG and go to the address of the flink field of SearchFilterHost.exe which is located in 0x862b5610. It contains the value 0x862e27d8 (ActiveProcessLink of calc.exe). Let's change it to the address of the ActiveProcessLink field of Taskmgr.exe, 0x862e4930:



We do the same with the blink of Taskmgr.exe which is located in 0x862e4930 and its value needs to be changed to 0x862b5610.

At this point we are supposed to be after the patching. Let's return control to the machine and see if that worked:



Success! We see that calc.exe is open, but the Processes tab of Task Manager doesn't think so.

Note: if we return now to the debugger and run "**!process 0 0**" again, we can see that calc.exe disappeared from the list, even though it is obviously running in the system. Therefore we can assume that the command "**!process**" uses this linked list mechanism as well.

Conclusion

We used the EPROCESS linked list mechanism in order to hide a process from the Task Manager. Obviously, this method isn't enough if you wish to hide a process, there are other methods to detect processes in the system (for example, iterating

through all of the running threads). Note that this method actually isn't enough even for Task Manager itself; Although calc.exe disappeared from the Processes tab, it is still visible through the Applications tab.

However, this little experiment does help us better understand the structure of the EPROESS linked list and how it may affect the system.