Mining the PFN Database for malware artifacts.

George M. Garner Jr.
President

GMG Systems, Inc.

Outline

- What is the PFN Database?
- Invalid PFN Entries.
- Invalid PFN Entries (cont.)
- Sparse PFN Database.
- PFN Entry (Detailed View).
- PFN Entry (Simplified View).
- An Example Application.

Outline (continued).

- Example (Original text).
- Example (English translation).
- Walking the Page Frame.
- Walking the Page Frame (cont.).
- Top Level Frame.
- The Owning Process.
- Switch to the Process Context.

Outline (continued).

- Using PTE Virtual Address.
- Using PTE Virtual Address (cont).
- PTE-To-VA Conversion.
- Hard coded in the Kernel: Where?
- Another Example…
- Browser Election Traffic.
- Browser Election (cont.).

Outline (continued).

- Eradicate the Infection.
- TDI Address Object.
- With a NTFS Pool Tag.
- In Deallocated Memory.
- So where did it come from?
- Finding the Allocating Code.
- Using the PFN Database.

Outline (continued).

- Locate the Module.
- An NDIS Packet Filter.
- Following the Leads.
- Thank you for your time.

What is the PFN Database?

- PFN database contains metadata about every physical page in the physical memory map.
- VMS OS has similar concept.
- Starts at PFN 0.
- Limit is nt! MmHighestPossiblePhysicalPage.
- Not nt!MmHighestPhysicalPage.

Invalid PFN Entries.

- Some PFN entries do not contain valid metadata:
  - PFN 0.
  - ISA or PCI option ROMs.
  - SMM memory.

Invalid PFN Entries (cont.)

– PCI reserved space.
  - 0xC0000000 – 0x100000000 (most Intel systems).
  - Some recent AMD systems reclaim memory above 0xC0000000.
  - Need to read the datasheet for your motherboard chipset.

Sparse PFN Database.

- Starting with Windows 7 SP1 PFN database is sparse.
  - PFN entries for PCI space may not be mapped into system virtual address space.
- Check nt!MiPfnBitMap to see if PFN entry is valid.

PFN Entry (Detailed View).

```
0: kd> dt -b nt!_MMPFN
   +0x000 u1               : __unnamed
      +0x000 Flink            : Uint4B
      +0x000 WsIndex          : Uint4B
      +0x000 Event            : Ptr32
      +0x000 ReadStatus       : Int4B
      +0x000 NextStackPfn     : _SINGLE_LIST_ENTRY
         +0x000 Next             : Ptr32
   +0x004 PteAddress       : Ptr32
   +0x008 u2               : __unnamed
      +0x000 Blink            : Uint4B
      +0x000 ShareCount       : Uint4B
   +0x00c u3               : __unnamed
      +0x000 e1               : _MMPFNENTRY
         +0x000 Modified         : Pos 0, 1 Bit
         +0x000 ReadInProgress   : Pos 1, 1 Bit
         +0x000 WriteInProgress  : Pos 2, 1 Bit
         +0x000 PrototypePte     : Pos 3, 1 Bit
         +0x000 PageColor        : Pos 4, 3 Bits
         +0x000 ParityError      : Pos 7, 1 Bit
         +0x000 PageLocation     : Pos 8, 3 Bits
         +0x000 RemovalRequested : Pos 11, 1 Bit
         +0x000 CacheAttribute   : Pos 12, 2 Bits
         +0x000 Rom              : Pos 14, 1 Bit
         +0x000 LockCharged      : Pos 15, 1 Bit
         +0x000 DontUse          : Pos 16, 16 Bits
      +0x000 e2               : __unnamed
         +0x000 ShortFlags       : Uint2B
         +0x002 ReferenceCount   : Uint2B
   +0x010 OriginalPte      : _MMPTE
      +0x000 u                : __unnamed
         +0x000 Long             : Uint4B
         +0x000 Hard             : _MMPTE_HARDWARE
            +0x000 Valid            : Pos 0, 1 Bit
            +0x000 Writable         : Pos 1, 1 Bit
            +0x000 Owner            : Pos 2, 1 Bit
```

PFN Entry (Simplified View).

0: kd> !pfn 1

PFN 00000001 at address 81469018

flink 000003F2 blink / share count 00000001 pteaddress C01DB504

reference count 0001 Cached color 0

restore pte 00AC3060 containing page 01CD3A Active

An Example.

PhysicalAddress: 0x07c35ca0
Hexadecimal display:

```
07c35ca0  8c 18 00 00 5b 33 32 6d-77 77 77 34 2e xx xx xx  ....[32mwww4.xxx
07c35cb0  73 xx xx xx xx 6e 65 74-1b 5b 30 6d 0d 0a 1b 5b  xxxx.net.[0m...[
07c35cc0  6d 0d 0a 1b 5b 32 31 3b-31 48 1b 5b 6d 1b 5b 6d  m...[21;1H.[m.[m
07c35cd0  1b 5b 31 6d bb b6 d3 ad-b9 e2 c1 d9 20 a1 f4 1b  .[1m........ ...
07c35ce0  5b 33 31 6d cb ae c4 be-c9 e7 c7 f8 1b 5b 33 37  [31m.........[37
07c35cf0  6d a1 f4 20 1b 5b 33 36-6d c9 cf cf df c8 cb ca  m.. .[36m.......
07c35d00  fd 20 1b 5b 31 6d 32 32-32 37 33 5b d7 ee b8 df  . .[1m22273[....
07c35d10  3a 20 32 34 37 37 37 5d-28 38 38 32 32 20 57 57  : 24777](8822 WW
07c35d20  57 20 47 55 45 53 54 29-1b 5b 6d 0d 0a 72 31 32  W GUEST).[m..r12
07c35d30  20 bb b6 d3 ad c4 fa ca-b9 d3 c3 73 73 68 b7 bd   ..........ssh..
07c35d40  ca bd b7 c3 ce ca a1 a3-b0 b4 20 5b 52 45 54 55  .......... [RETU
07c35d50  52 4e 5d 20 bc cc d0 f8-00 5f 5c 20 0c b6 5a 00  RN] ....._\ ..Z.
```

Example (Original text).

Chinese:

[32mwww4.xxxxxxx.net [0m

[m

[21;1H [m [m [1m 欢迎光临 ◆ [31m XXXX [37m◆ [36m 上线人数 [1m22273[最高: 24777](8822 WWW GUEST) [m

r12 欢迎您使用ssh 方式访问。按 [RETURN] 继续 _\

Example (English translation).

- English translation.

[32mwww4.xxxxxxx.net [0m

[M

[21; 1H [m [m [1m Welcome ◆ [31m Xxxxx community [37m ◆ [36m line on the number of [1m22273 [Maximum: 24777]

(8822 WWW GUEST) [m

r12 Thank you for using ssh access. Press [RETURN] to continue _ \

Walking the Page Frame...

PhysicalAddress: 0x**07c35**ca0

```
0: kd> !pfn 07c35
    PFN 00007C35 at address 815234F8
    flink       000003C7  blink / share count 00000001
  pteaddress C00054A4
    reference count 0001   Cached     color 0
    restore pte 00000080
    containing page    011304
    Active     M
    Modified
```

Walking the Page Frame (cont.)

```
0: kd> !pfn 011304
    PFN 00011304 at address 81605860
    flink       0000009F  blink / share count 0000005A
   pteaddress C0300014
    reference count 0001   Cached     color 0
    restore pte 00000080
  containing page      00F1DD
  Active     M
    Modified
```

Top Level Frame.

```
0: kd> !pfn 00F1DD
    PFN 0000F1DD at address 815D3CB8
    flink  8617BC88  blink / share count 0000001E
  pteaddress C0300C00
    reference count 0001   Cached     color 0
    restore pte 00000080  containing page        00F1DD
  Active     M

    Modified
```

The Owning Process.

```
0: kd> !process 8617BC88
PROCESS 8617bc88  SessionId: 0  Cid: 08ac    Peb:
  7ffd3000  ParentCid: 0ed0
    DirBase: 0f1dd000  ObjectTable: e623f5e8
  HandleCount: 198.
    Image: fterm.exe
    VadRoot 8527a2c8 Vads 207 Clone 0 Private 2041.
  Modified 8034. Locked 0.
    DeviceMap e3fa28f0
    Token                             e7922610
    ElapsedTime                       2 Days
  16:22:13.173
    UserTime                          00:00:44.046
```

Switch to the Process Context.

```
0: kd> .process /p /r 8617BC88
Implicit process is now 8617bc88
Loading User Symbols
...............................
  ..................
```

Using PTE Virtual Address.

```
0: kd> !pfn 07c35
    PFN 00007C35 at address 815234F8
    flink       000003C7  blink / share
  count 00000001  pteaddress C00054A4
    reference count 0001   Cached
  color 0
    restore pte 00000080
  containing page          011304
  Active      M
    Modified
```

Using PTE Virtual Address (cont).

```
0: kd> ?0`C00054A4<<a
Evaluate expression: 3298557071360 =
  00000300`01529000
```

```
0: kd> db 1529000 + ca0
01529ca0  8c 18 00 00 5b 33 32 6d-77 77 77 34 2e xx xx xx  ....[32mwww4.xxx
01529cb0  xx xx xx xx 2e 6e 65 74-1b 5b 30 6d 0d 0a 1b 5b  xxxx.net.[0m...[
01529cc0  6d 0d 0a 1b 5b 32 31 3b-31 48 1b 5b 6d 1b 5b 6d  m...[21;1H.[m.[m
01529cd0  1b 5b 31 6d bb b6 d3 ad-b9 e2 c1 d9 20 a1 f4 1b  .[1m........ ...
01529ce0  5b 33 31 6d cb ae c4 be-c9 e7 c7 f8 1b 5b 33 37  [31m.........[37
01529cf0  6d a1 f4 20 1b 5b 33 36-6d c9 cf cf df c8 cb ca  m.. .[36m.......
01529d00  fd 20 1b 5b 31 6d 32 32-32 37 33 5b d7 ee b8 df  . .[1m22273[....
01529d10  3a 20 32 34 37 37 37 5d-28 38 38 32 32 20 57 57  : 24777](8822 WW
01529d20  57 20 47 55 45 53 54 29-1b 5b 6d 0d 0a 72 31 32  W GUEST).[m..r12
01529d30  20 bb b6 d3 ad c4 fa ca-b9 d3 c3 73 73 68 b7 bd   ..........ssh..
01529d40  ca bd b7 c3 ce ca a1 a3-b0 b4 20 5b 52 45 54 55  .......... [RETU
01529d50  52 4e 5d 20 bc cc d0 f8-00 5f 5c 20 0c b6 5a 00  RN] ....._\ ..Z.
```

PTE-To-VA Conversion.

- X86: ?0`C00054A4<<a
- X86 PAE: ?0`C04C70E0<<9
- X64:

```
1: kd> ?0`FFFFF6FC40022C58<<19
Evaluate expression: -5404271763280363 52 = f8800458`b0000000
1: kd> ?0`f8800458`b0000000>>10
Evaluate expression: 273228712423424 = 0000f880`0458b000
1: kd> ?0`0000f880`0458b000 | ffff000000000000
Evaluate expression: -8246264287232 = ffff f880`0458b000
```

Hard coded in the Kernel: Where?

- nt!MmGetVirtualForPhysical
- nt!MmAccessFault
- OS loader.

Another Example.

- A laptop was actively scanning external networks for TCP port 445.
- Physical memory was acquired from the infected box and inline hooks were found consistent with Conficker:
  - DNSAPI!DnsQuery_W
  - DNSAPI!DnsQuery_UTF8
  - ntdll!NtQueryInformationProcess

Browser Election Traffic

- Browser election traffic included a file listing of the incident response CD.

```
kd> db 84a99000 L6a0
84a99000  00 00 d4 0a 41 74 74 76-98 06 00 00 b4 a6 38 86  ....Attv......8.
84a99010  00 00 00 00 00 00 00 00-0e 00 cf 00 cf 00 00 00  ................
84a99020  00 00 00 00 00 00 00 00-00 00 00 00 ff ff ff ff  ................
84a99030  ff ff 00 60 b0 20 f2 8e-08 00 45 00 00 cf bb b9  ...`. ....E.....
84a99040  00 00 40 11 b6 2a 2f 8f-54 1d 2f 8f 54 ff 00 8a  ..@..*/.T./.T...
84a99050  00 8a 00 bb 61 0b 11 02-00 00 2f 8f 54 1d 00 8a  ....a...../.T...
84a99060  00 a5 00 00 20 46 48 46-43 46 45 46 41 45 49 44  .... FHFCFEFAEID
84a99070  41 45 4e 45 4e 43 41 43-41 43 41 43 41 43 41 43  AENENCACACACACAC
84a99080  41 43 41 43 41 00 20 46-46 46 44 46 43 46 45 46  ACACA. FFFDFCFEF
84a99090  41 44 41 44 42 43 41 43-41 43 41 43 41 43 41 43  ADADBCACACACACAC
84a990a0  41 43 41 43 41 42 4e 00-ff 53 4d 42 25 00 00 00  ACACABN..SMB%...
84a990b0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
84a990c0  00 00 00 00 00 00 00 00-11 00 00 0b 00 00 00 00  ................
84a990d0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 0b  ................
84a990e0  00 56 00 03 00 01 00 01-00 02 00 1c 00 5c 4d 41  .V...........\MA
84a990f0  49 4c 53 4c 4f 54 5c 42-52 4f 57 53 45 00 02 00  ILSLOT\BROWSE...
84a99100  57 52 54 50 48 30 4d 4d-00 30 4d 4d 00 00 00 00  WRTPH0MM.0MM....
```

Browser Election (cont.)

```
[…]

84a991f0  00 00 20 00 00 00 9d 3b-15 00 68 70 76 69 65 77  .. ....;..hpview
84a99200  2f 6d 73 76 63 70 38 30-2e 64 6c 6c 50 4b 01 02  /msvcp80.dllPK..
84a99210  14 00 14 00 01 00 08 00-72 a8 81 35 0f 79 91 18  ........r..5.y..
84a99220  e5 dc 04 00 00 90 09 00-12 00 00 00 00 00 00 00  ................
84a99230  00 00 20 00 00 00 af 7f-17 00 68 70 76 69 65 77  .. .......hpview
84a99240  2f 6d 73 76 63 72 38 30-2e 64 6c 6c 50 4b 01 02  /msvcr80.dllPK..
84a99250  14 00 14 00 01 00 08 00-a9 90 12 39 d5 39 ad a9  ...........9.9..
84a99260  0f ed 03 00 00 00 0b 00-0d 00 00 00 00 00 00 00  ................
84a99270  00 00 20 00 00 00 c4 5c-1c 00 68 70 76 69 65 77  .. ....\..hpview
84a99280  2f 6e 63 2e 65 78 65 50-4b 01 02 14 00 14 00 01  /nc.exePK.......
84a99290  00 08 00 f7 91 12 39 02-5c 11 bf 83 01 02 00 00  ......9.\.......
84a992a0  40 05 00 16 00 00 00 00-00 00 00 00 00 20 00 00  @............ ..
84a992b0  00 fe 49 20 00 68 70 76-69 65 77 2f 76 6f 6c 75  ..I .hpview/volu
84a992c0  6d 65 5f 64 75 6d 70 2e-65 78 65 50 4b 01 02 14  me_dump.exePK...
```

Eradicate the Infection.

- So the laptop hard drive was wiped and re-imaged from factory media.
- The machine then was connected to the network.
- After 10 minutes physical memory was reacquired from the reimaged laptop.
- Result was a nice clean box with no IOC.

Or So We Thought.

- Except for one.
- KnTList log entry:

```
Warning: TDI address object
  0x859516F0 has an unexpected
  allocation pool tag:
  0x80562858!
```

TDI Address Object.

- A UDP address object.

```
Address Object: 0x859516F0 (55516f0)
Local Address: 0x0:d0ce 0.0.0.0:52944
Protocol: 17 MCastIF: 0x0
Flags1: 0x48  Flags2: 0x4
AssociatedConnections: { -:-} {-:-}
ProcessId: 0x4c0 svchost.exe \Device
\HarddiskVolume1\WINDOWS
\system32\svchost.exe
BindTimestamp: 0x1c90cxxxxx26090
20xx-09-01 15:50:37Z
```

With a NTFS Pool Tag.

```
kd> dds /c1 859516f0-10 L5e
859516e0  002f0008
859516e4  6e66744e <- 'Ntfn' pool allocation tag
859516e8  86fa1180 <-Flink
859516ec  80562858 <-Blink
                     nt!NonPagedPoolDescriptor+0x198
859516f0  85a1b008 <- 'Attv' pool allocation tag
859516f4  00000000
859517d0  00000000
[…]
859517d4  00000000
859517d8  aaccc7db tcpip!UDPSend
859517dc  00000000
```

In Deallocated Memory.

- Perhaps you might think that this is something old, that has been deleted and partially overwritten.
- However, the TDI object is in the tcpip!AddrObjTable.

```
kd> !pool 859516f0
Pool page 859516f0 region is Nonpaged pool
[...]
*859516e0 size:  178 previous size:   40  (Free)
  *Ntfn
      Pooltag Ntfn : SCB_NONPAGED, Binary : ntfs.sys
```

So where did it come from?

- The ‘Ntfn’ pool tag is probably not going to help us.
- The block of memory was in use by the NTFS subsystem and then freed. It keeps the ‘Ntfn’ pool tag until it is reused.
- Someone is reusing the block of memory without reallocating it.
- Think of it as “stolen” memory.

Finding the Allocating Code.

- How about the embedded buffer with the '**Attv**' pool tag?
- So we searched the memory dump for that pool tag and found a physical page with code that references this pool tag.

```
kd> up 8D2Cab1
08d2cab1 894118          mov     dword ptr [ecx+18h],eax
08d2cab4 8bb3e8030000    mov     esi,dword ptr [ebx+3E8h]
08d2caba 6841747476      push    76747441h  ;  'Attv'
08d2cabf 8945e4          mov     dword ptr [ebp-1Ch],eax
08d2cac2 83c604          add     esi,4
08d2cac5 56              push    esi
08d2cac6 8d8538ffffff    lea     eax,[ebp-0C8h]
08d2cacc 50              push    eax
```

Using the PFN Database.

- Convert the PTE address to a virtual address.

```
kd> !pfn 8d2c
    PFN 00008D2C at address 81AE6C20
    flink       00000000  blink / share count 00000001
  pteaddress C03DB7C0
    reference count 0001   Cached     color 0
    restore pte 00000060  containing page        0015EF
  Active     M
    Modified

kd> ?0`C03DB7C0<<a
Evaluate expression: 3302676692992 = 00000300`f6df0000
```

Locate the Module.

- Locate the containing module by displaying the diassembly.

```
kd> u f6df0000+ab1
agnfilt+0x5ab1:
f6df0ab1 894118          mov     dword ptr [ecx+18h],eax
f6df0ab4 8bb3e8030000    mov     esi,dword ptr [ebx+3E8h]
f6df0aba 6841747476      push    76747441h ; 'Attv'
f6df0abf 8945e4          mov     dword ptr [ebp-1Ch],eax
f6df0ac2 83c604          add     esi,4
f6df0ac5 56              push    esi
f6df0ac6 8d8538ffffff    lea     eax,[ebp-0C8h]
f6df0acc 50              push    eax
```

An NDIS Packet Filter.

- Turns out the block of memory was allocated by a network packet filter that is part of a commercial business Internet services and VPN client software product:
- The packet filter needs to inspect content before it is transmitted over the wire.
- We searched the packets waiting to be sent and found suspicious browser election traffic.

Following the Leads.

- We started with a UDP address object in deallocated memory.
- And the purported owning process (“svchost.exe”) of the UDP object.
- Thanks to the PFN database we now have an NDIS packet filter to add to our list of suspects.
- And we have some investigative leads to follow up on.

Thank you for your time.

Contact:

http://www.gmgsystemsinc.com/knttools/
gmgarner (at) gmgsystemsinc (dot) com.