Analyzing SMC Malware

Extraction and Analysis of Hidden Code

The Sample

Property	Value
File Name	E:\Projects\Misc\CTF\Nullcon\2014\HackIM\RE\Hard_Stealth_Downloader\L...
File Type	Portable Executable 32
File Info	Microsoft Visual C++ 8
File Size	81.00 KB (82944 bytes)
PE Size	81.00 KB (82944 bytes)
Created	Tuesday 21 January 2014, 16.06.26
Modified	
Accessed	
MD5	
SHA-1	

Module Name	Imports	OFTs	TimeDateStamp	ForwarderChain	Name RVA	FTs (IAT)
00008304	N/A	00008194	00008198	0000819C	000081A0	000081A4
szAnsi	(nFunctions)	Dword	Dword	Dword	Dword	Dword
PSAPI.DLL	2	000096D0	00000000	00000000	00009704	00008100
KERNEL32.dll	63	000095D0	00000000	00000000	00009B9A	00008000

	FTs (IAT)	Hint	Name
Dword	Dword	Word	szAnsi
000096F4	000096F4	0006	EnumProcesses
000096DC	000096DC	000F	GetModuleFileNameExA

Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
.text	0000669A	00001000	00006800	00000400	00000000	00000000	0000	0000	60000020
.rdata	00001BA8	00008000	00001C00	00006C00	00000000	00000000	0000	0000	40000040
.data	0000D6A0	0000A000	0000BC00	00008800	00000000	00000000	0000	0000	C0000040

Investigation – 1/2 Debugger Detection

IDA - E:\Projects\Misc\CTF\Nullcon\2014\HackIM\RE\Hard_Stealth_Downloader\Loader.exe

File Edit Jump Search View Debugger Options Windows Help

Functions window

Function name

sub_401000
sub_401080
WinMain(x,x,x,x)
sub_401110
sub_401250
sub_401290
GetModuleFileNameExA
EnumProcesses
_LocaleUpdate::_LocaleUpdate(localeinfo_st
__tolower_l
_tolower
_strlen
_memset
sub_40158A
__alloca_probe
_strstr
_fast_error_exit
___tmainCRTStartup
$LN33
_CPtoLCID

IDA View-A | Hex View-A | Structures | Enums | Imports | Exports

```
; Attributes: bp-based frame

; int __stdcall W                                   evInstance, LPSTR lpCmdLine, int nShowCmd)
_WinMain@16 proc

hInstance= dword
hPrevInstance= dw
lpCmdLine= dword
nShowCmd= dword p

push    ebp
mov     ebp, esp
call    sub_40125
call    sub_40111
xor     eax, eax
pop     ebp
retn    10h
_WinMain@16 endp
```

```
; Attributes: bp-based frame

sub_401250 proc near

Wow64Process= dword ptr -4

push    ebp
mov     ebp, esp
push    ecx
mov     [ebp+Wow64Process], 0
lea     eax, [ebp+Wow64Process]
push    eax             ; Wow64Process
call    ds:GetCurrentProcess
push    eax             ; hProcess
call    ds:IsWow64Process
cmp     [ebp+Wow64Process], 0
jz      short loc_40127A
push    0               ; uExitCode
call    ds:ExitProcess

loc_40127A:
call    ds:IsDebuggerPresent
test    eax, eax
jz      short loc_40128C
push    0               ; uExitCode
call    ds:ExitProcess

loc_40128C:
mov     esp, ebp
pop     ebp
retn
sub_401250 endp
```

Graph overview

100.00% (-55,96) (243,242) 00000

Output window

```
Can not set debug privilege: Not all privileges or groups referenced are
LoadLibrary(C:\Program Files\IDA\plugins\zynamics_binexport_5.plw) error
C:\Program Files\IDA\plugins\zynamics_binexport_5.plw: can't load file
Using FLIRT signature: Microsoft VisualC 2-9/net runtime
Propagating type information...
Function argument information has been propagated
The initial autoanalysis has been finished.
```

Python

AU: idle | Down | Disk: 11GB | Click on node title to select/drag it; DblClick on edge to follow it;

Start | 4:19 PM 1/21/2014

Investigation – 2/2 Process Injection

IDA - E:\Projects\Misc\CTF\Nullcon\2014\HackIM\RE\Hard_Stealth_Downloader\Loader

File Edit Jump Search View Debugger Options Windows Help

Functions window

Function name

- sub_401000
- sub_401080
- WinMain(x,x,x,x)
- sub_401110
- sub_401250
- sub_401290
- GetModuleFileNameExA
- EnumProcesses
- _LocaleUpdate::_LocaleUpdate(localeinfo_st
- __tolower_l
- _tolower
- _strlen
- _memset
- sub_40158A
- __alloca_probe
- _strstr
- _fast_error_exit
- ___tmainCRTStartup
- $LN33

IDA View-A | Hex View-A

```
lpStartAddress= dword ptr -0Ch
var_8= dword ptr -8
ThreadId= dword ptr -4
hProcess= dword ptr  8
lpBuffer= dword ptr  0Ch
nSize= dword ptr  10h

push    ebp
mov     ebp, esp
sub     esp, 0Ch
push    40h             ; flProtect
push    3000h           ; flAllocationType
mov     eax, [ebp+nSize]
add     eax, 1
push    eax             ; dwSize
push    0               ; lpAddress
mov     ecx, [ebp+hProcess]
push    ecx             ; hProcess
call    ds:VirtualAllocEx
mov     [ebp+lpStartAddress], eax
cmp     [ebp+lpStartAddress], 0
jnz     short loc_40102D
xor     eax, eax
jmp     short loc_401073

loc_40102D:
lea     edx, [ebp+ThreadId]
push    edx             ; lpNumberOfBytesWritten
mov     eax, [ebp+nSize]
push    eax             ; nSize
mov     ecx, [ebp+lpBuffer]
push    ecx             ; lpBuffer
mov     edx, [ebp+lpStartAddress]
push    edx             ; lpBaseAddress
mov     eax, [ebp+hProcess]
push    eax             ; hProcess
call    ds:WriteProcessMemory
lea     ecx, [ebp+ThreadId]
push    ecx             ; lpThreadId
push    0               ; dwCreationFlags
push    0               ; lpParameter
mov     edx, [ebp+lpStartAddress]
push    edx             ; lpStartAddress
push    0               ; dwStackSize
push    0               ; lpThreadAttributes
mov     eax, [ebp+hProcess]
push    eax             ; hProcess
call    ds:CreateRemoteThread
mov     [ebp+var_8], eax
cmp     [ebp+var_8], 0
jnz     short loc_40106E
xor     eax, eax
```

Output window

```
Can not set debug privilege: Not all privileges or groups referenced are a

LoadLibrary(C:\Program Files\IDA\plugins\zynamics_binexport_5.plw) error:
C:\Program Files\IDA\plugins\zynamics_binexport_5.plw: can't load file
Using FLIRT signature: Microsoft VisualC 2-9/net runtime
Propagating type information...
Function argument information has been propagated
The initial autoanalysis has been finished.
```

Graph overview

AU: idle | Down | Disk: 11GB | Click on node title to select/drag it; DblClick on edge to follow it; Wh

Start | 4:22 PM 1/21/2014

Investigation – Target Process

```
push    ebp
mov     ebp, esp
mov     eax, 1144h
call    __alloca_probe
mov     eax, dword_414F10
xor     eax, ebp
mov     [ebp+var_8], eax
mov     [ebp+var_1], 0
lea     eax, [ebp+cbNeeded]
push    eax                 ; lp
push    1000h               ; cb
lea     ecx, [ebp+dwProcessI
push    ecx                 ; lp
call    EnumProcesses
test    eax, eax
jz      loc_40123C
```

```
add     esp, 0Ch
push    124h                ; nSize
lea     edx, [ebp+Filename]
push    edx                 ; lpFilename
push    0                   ; hModule
mov     eax, [ebp+hProcess]
push    eax                 ; hProcess
call    GetModuleFileNameExA
test    eax, eax
jnz     short loc_4011EC
```

```
loc_4011EC:                 ; "explorer.exe"
push    offset aExplorer_exe
lea     ecx, [ebp+Filename]
push    ecx                 ; char *
call    sub_401080
add     esp, 4
push    eax                 ; char *
call    sub_401290
```

Extraction of Injected Code

```
E:\Projects\OSS\RandomCode\Malware\HiDump>ruby HiDump.rb -t E:\Projects\Misc\CTF\Nullcon\2014\HackIM
\RE\Hard_Stealth_Downloader\Loader.exe -v -d C:\tmp -x
[+] Starting Target with CREATE_SUSPENDED flag (E:\Projects\Misc\CTF\Nullcon\2014\HackIM\RE\Hard_Ste
alth_Downloader\Loader.exe)
[+] Process created with pid: 3628
[+] Injecting monitor dll
[+] Preparing monitor dll with config
[*] Configured monitor dll path: C:\Users\DEVELO~1\AppData\Local\Temp\hid-monitor20140121-604-18xpwg
c.dll
[+] Waiting
[+] Resuming execution of target
[+] Waiting for target to finish execution
[+] Finished
```

Injected Code

```
seg000:00000000 seg000          segment byte public 'CODE' use32
seg000:00000000                 assume cs:seg000
seg000:00000000                 assume es:nothing, ss:nothing, ds:
seg000:00000000                 db 0B0h ; ¦
seg000:00000001                 db  6Ch ; l
seg000:00000002                 db 0B9h ; ¦
seg000:00000003                 db 0F1h ; ±
seg000:00000004                 db 0AEh ; «
seg000:00000005                 db    0
seg000:00000006                 db    0
seg000:00000007                 db 0EBh ; d
seg000:00000008                 db  0Ah
seg000:00000009                 db  5Eh ; ^
seg000:0000000A                 db  89h ; ë
seg000:0000000B                 db 0F3h ; =
seg000:0000000C                 db  30h ; 0
seg000:0000000D                 db    6
seg000:0000000E                 db  46h ; F
seg000:0000000F                 db 0E2h ; G
seg000:00000010                 db 0FBh ; v
seg000:00000011                 db 0FFh
seg000:00000012                 db 0D3h ; +
seg000:00000013                 db 0E8h ; F
seg000:00000014                 db 0F1h ; ±
```

Injected Code

```
seg000:00000000 ; Segment type: Pure code
seg000:00000000 seg000          segment byte public 'CODE' use32
seg000:00000000                 assume cs:seg000
seg000:00000000                 assume es:nothing, ss:nothing, ds:not
seg000:00000000                 mov     al, 6Ch ; 'l'
seg000:00000002                 mov     ecx, 0AEF1h
seg000:00000007                 jmp     short loc_13
seg000:00000009
seg000:00000009 ; =============== S U B R O U T I N E ===============
seg000:00000009
seg000:00000009
seg000:00000009 sub_9           proc near               ; CODE XREF:
seg000:00000009                 pop     esi
seg000:0000000A                 mov     ebx, esi
seg000:0000000C
seg000:0000000C loc_C:                                  ; CODE XREF:
seg000:0000000C                 xor     [esi], al
seg000:0000000E                 inc     esi
seg000:0000000F                 loop    loc_C
seg000:00000011                 call    ebx
seg000:00000013
seg000:00000013 loc_13:                                 ; CODE XREF:
seg000:00000013                 call    sub_9
seg000:00000018                 fsubr   st(1), st
```

Dynamic Analysis

- Impossible to understand SMC statically !
- We need to execute the code in memory while tracing through it.

- **Approach**
  - Ruby script that reads extracted code form file and executes with 0xcc prepended.
  - Debugger is attached to ruby.exe process.
  - On int3 hit, debugger gets control.
  - Continue with single-step based tracing.

```
print "[*] Press enter to load and execute shellcode .. "
$stdin.gets

process = ::Metasm::WinOS::Process.new(::Metasm::WinAPI.getcurrentprocessid)
::Metasm::WinOS.inject_run_shellcode(process, "\xcc" + File.binread(sc_path))

puts "[*] Shellcode loaded and executed"
loop do
    puts "Sleeping .... "
    sleep 5
end
```

SMC – Decoder Loops

```
debug020:003F000D
debug020:003F000D loc_3F000D:                          ; CODE XREF: debug020:003F0010↓j
debug020:003F000D xor     [esi], al
debug020:003F000F inc     esi
EIP debug020:003F0010 loop    loc_3F000D
debug020:003F0012 call    ebx
```

XOR Decoder

```
debug020:003F0080 ; ---------------------------------------
debug020:003F0080 fnstenv byte ptr [esp-0Ch]
debug020:003F0084 pop     ebp
EBP debug020:003F0085 sub     ecx, ecx
debug020:003F0087 mov     cx, 2B9Eh
debug020:003F008B
debug020:003F008B loc_3F008B:
EIP debug020:003F008B xor     [ebp+19h], edx
debug020:003F008E sub     ebp, 0FFFFFFFCh
debug020:003F0091 add     edx, [ebp+15h]
debug020:003F0093 adc     eax, 4BBFF5E2h
debug020:003F0094 loop    loc_3F008B
debug020:003F0098 and     al, 0A1h
```

Shikata Ga Nai Decoder

Found Embedded PE

```
debug022:003400F9 xor     ecx, ecx
debug022:003400FB mov     cx, 2B81h
debug022:003400FF
debug022:003400FF loc_3400FF:                          ; CODE XREF: debug022:00340108↓j
debug022:003400FF sub     eax, 0FFFFFFFCh
debug022:00340102 xor     [eax+15h], ebp
debug022:00340105 add     ebp, [eax+15h]
debug022:00340108 loop    loc_3400FF
debug022:00340108 ; ---------------------------------------------------------------------------
debug022:0034010A db 4Dh                               ; Start of PE MZ Header
debug022:0034010B db 5Ah
debug022:0034010C db 0E8h
debug022:0034010D db    0
debug022:0034010E db    0
debug022:0034010F db    0
debug022:00340110 db    0
debug022:00340111 db 5Bh
debug022:00340112 db 52h
debug022:00340113 db 45h
debug022:00340114 db 55h
```

.. after spending an eternity single-stepping instructions

Memory Dump of Embedded PE

```
debug022:003400FF
debug022:003400FF loc_3400FF:                          ; CODE XREF: debug022:00340108↓j
debug022:003400FF sub     eax, 0FFFFFFFCh
debug022:00340102 xor     [eax+15h], ebp
debug022:00340105 add     ebp, [eax+15h]
debug022:00340108 loop    loc_3400FF
debug022:00340108 ; ---------------------------------------------------------------------
debug022:0034010A db 4Dh                               ; Start of PE MZ Header
debug022:0034010B db 5Ah
debug022:0034010C db 0E8h
debug022:0034010D db    0
debug022:0034010E db    0
```

Please enter a string

Please enter a WinDbg debugger command: .writemem C:\tmp\e.mem 0034010a 00400000

OK Cancel

We have a DLL being Injected

```
; BOOL __stdcall DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
_DllMain@12 proc near

var_18= dword ptr -18h
var_10= dword ptr -10h
var_8= dword ptr -8
var_4= dword ptr -4
hinstDLL= dword ptr  8
fdwReason= dword ptr  0Ch
lpvReserved= dword ptr  10h

push    ebp
mov     ebp, esp
push    0FFFFFFFEh
push    offset unk_100095A0
push    offset __except_handler4
mov     eax, large fs:0
```

Core Logic of Injected Code

```
sub     esp, 28h
mov     eax, dword_1000A000
xor     eax, esp
mov     [esp+28h+var_4], eax
xor     eax, eax
mov     dword ptr [esp+28h+First], eax
mov     [esp+28h+var_10], eax
mov     [esp+28h+var_C], eax
mov     [esp+28h+var_8], eax
lea     eax, [esp+28h+nSize]
push    eax             ; nSize
lea     ecx, [esp+2Ch+First]
push    ecx             ; lpBuffer
mov     [esp+30h+nSize], 0Fh
call    ds:GetComputerNameA
push    offset Srch     ; "NULLCON2014"
lea     edx, [esp+2Ch+First]
push    edx             ; lpFirst
call    ds:StrStrA
test    eax, eax
jz      short loc 100013B8
```

Core Logic of Injected Code

```
xor     eax, eax
mov     [esp+28h+var_28], eax
mov     [esp+28h+var_24], eax
mov     [esp+28h+var_20], eax
mov     [esp+28h+var_1C], eax
lea     eax, [esp+28h+var_28]
push    eax                     ; lpSystemTime
call    ds:GetSystemTime
mov     eax, 3
cmp     word ptr [esp+28h+var_24+2], ax
jnz     short loc_100013AE
```

```
cmp     word ptr [esp+28h+var_28+2], ax
jnz     short loc_100013AE
```

```
mov     ecx, 7F1h
cmp     word ptr [esp+28h+var_28], cx
jnz     short loc_100013AE
```