Win-Trojan/Patched.H 분석 따라하기

1. 개 요

Win-Trojan/Patched.H는 Win-Trojan/Agent.52736.FU가 생성한 %SYSTEM%₩Sysldt.dll(Win-Trojan/Hupigon.56448)를 자동으로 로딩하도록 윈도우 쉘(Shell)인 Explorer.exe에 삽입한 바이러스 코드에 대한 진단명입니다.

백신에서 %SYSTEM%₩Sysldt.dll(Win-Trojan/Hupigon.56448)만 삭제할 경우 발생했던 문제인 재 부팅 시 Explorer.exe에서 에러 및 실행불가로 인해 바탕화면이 뜨지 않는 현상의 원인에 대해서 알아 보겠습니다.

2. Explorer.exe 감염

Explorer.exe를 감염시키는 역할을 하는 파일은 %SYSTEM%₩Sysldt.dll이고 Explorer.exe의 감염 전과 후를 살펴 보면 아래와 같습니다.

* 감염 전

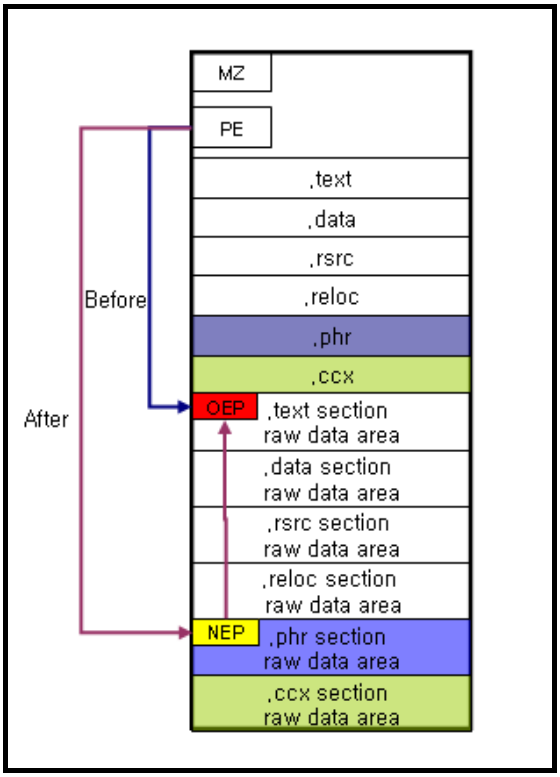
File: E:₩Personal₩Security₩Research₩Trojan.Patched.H₩Explorer Patch₩No_explorer.ex
Size: 1030144 bytes / MD5: 73A31B42AC0E198C4F4F62073C9EAD34 / CRC32: E8BFE6DC
Number of Sections : 00000004
Size of Code : 00044800
Entry Point : 0001E24E
Base of Code : 00001000
Base of Data : 00044000
Image Base : 01000000
Section Alignment : 00001000
File Alignment : 00000200
Size of Image : 000FE000
Size of Headers : 00000400

No	Name	VirtualSize	VirtualOffset	RawSize	RawOffset	Characteri...
01	.text	00044689	00001000	00044800	00000400	60000020
02	.data	00001D90	00046000	00001800	00044C00	C0000040
03	.rsrc	000B1B30	00048000	000B1C00	00046400	40000040
04	.reloc	000036DC	000FA000	00003800	000F8000	42000040

* 감염 후

File: E:₩Personal₩Security₩Research₩Trojan.Patched.H₩Explorer Patch₩Yes_explorer.ex
Size: 1038336 bytes / MD5: 177005FC1C678BA0BC85B0083971AC72 / CRC32: C97FA90D
Number of Sections : 00000006
Size of Code : 00046800
Entry Point : 00100000
Base of Code : 00001000
Base of Data : 00044000
Image Base : 01000000
Section Alignment : 00001000
File Alignment : 00000200
Size of Image : 00106000
Size of Headers : 00000400

No	Name	VirtualSize	VirtualOffset	RawSize	RawOffset	Characteri...
01	.text	00044689	00001000	00044800	00000400	60000020
02	.data	00001D90	00046000	00001800	00044C00	C0000040
03	.rsrc	000B1B30	00048000	000B1C00	00046400	40000040
04	.reloc	000036DC	000FA000	00003800	000F8000	42000040
05	.phr	00001000	000FE000	00001000	000FB800	E0000020
06	.ccx	00001000	000FF000	00001000	000FC800	C0000040



위 그림을 보면 Win-Trojan/Patched.H 역시 전형적인 후위 형 바이러스임을 알 수가 있습니다.

Win-Trojan/Patched.H(이하, Patched.H)에 의해서 감염된 Explorer.exe를 Olly DBG로 로딩시키면 Patched.H의 NEP로 이동함을 알 수가 있습니다. 그리고 Patched.H의 바이러스 코드는 PushAD ~ PopAD 블록으로 구성되어 있습니다.

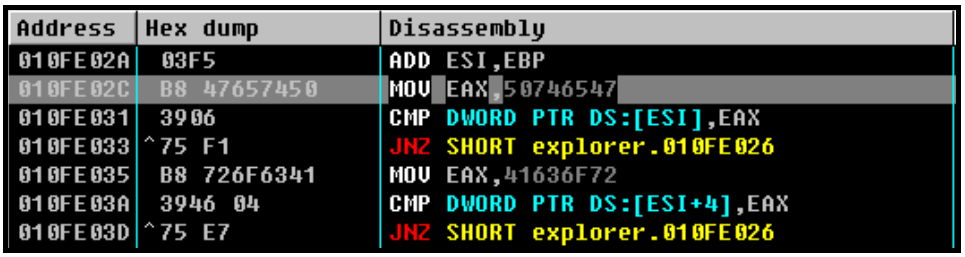
```
010FE000  60              PUSHAD
010FE001  64:A1 30000000  MOV EAX,DWORD PTR FS:[30]
010FE007  8B40 0C         MOV EAX,DWORD PTR DS:[EAX+C]
010FE00A  8B70 1C         MOV ESI,DWORD PTR DS:[EAX+1C]
010FE00D  AD              LODS DWORD PTR DS:[ESI]
010FE00E  8B40 08         MOV EAX,DWORD PTR DS:[EAX+8]
```

DS:[EAX+8]에는 메모리에 로딩된 kernel32.dll의 Base Address가 저장되어 있으며 시스템 및 OS의 버전에 따라서 조금씩 다른데 여기서는 0x7C800000h입니다.

```
Address   Hex dump     Disassembly
010FE015  8B45 3C      MOV EAX,DWORD PTR SS:[EBP+3C]
010FE018  8B5405 78    MOV EDX,DWORD PTR SS:[EBP+EAX+78]
010FE01C  03D5         ADD EDX,EBP
010FE01E  8B4A 18      MOV ECX,DWORD PTR DS:[EDX+18]
010FE021  8B5A 20      MOV EBX,DWORD PTR DS:[EDX+20]
010FE024  03DD         ADD EBX,EBP
010FE026  49           DEC ECX
010FE027  8B348B       MOV ESI,DWORD PTR DS:[EBX+ECX*4]
```

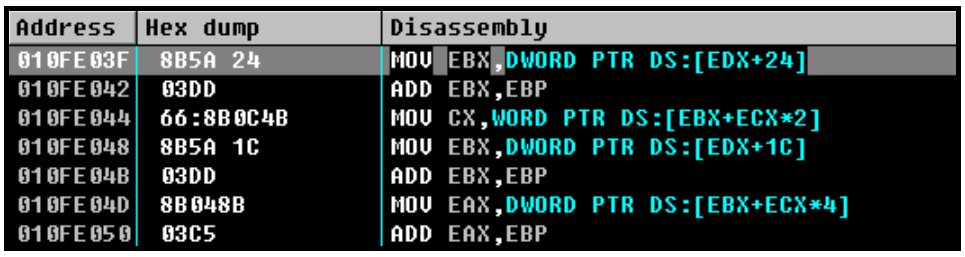
[EBP+3C]에는 kernel32.dll의 PE헤더 위치를 가리키는 포인터 주소가 저장되어 있고 [EBP+EAX+78]에는 Export Directory RVA가 저장되어 있습니다. 그리고 [EDX+18]은 Kernel32.dll이 가지고 있는 함수의 총 개수를 의미하는 Number of Functions의 값이, [EDX+20]에는 각 Function의 Name에 대한 포인터인 Name Pointer Table RVA가 저장되어 있습니다.

[EBX+ECX*4]에는 Name Pointer Table에서 마지막 함수인 lstrlenW의 포인터 주소가 저장되어 있는데 역순으로 검색하면서 GetProcAddress()함수의 포인터 주소를 찾기 위해서 입니다.



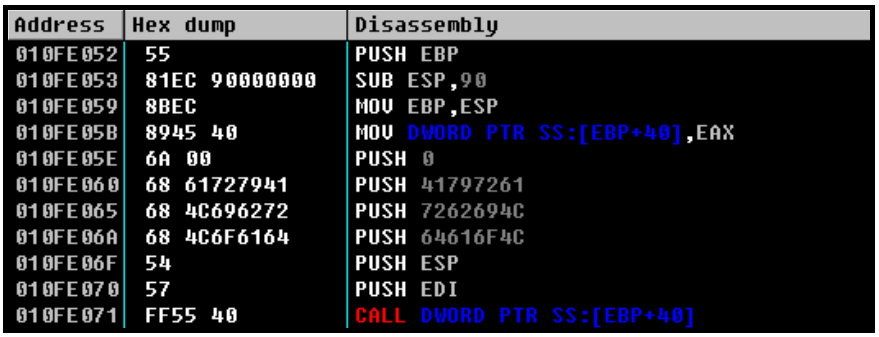
Address	Hex dump	Disassembly
010FE02A	03F5	ADD ESI,EBP
010FE02C	B8 47657450	MOV EAX,50746547
010FE031	3906	CMP DWORD PTR DS:[ESI],EAX
010FE033	^75 F1	JNZ SHORT explorer.010FE026
010FE035	B8 726F6341	MOV EAX,41636F72
010FE03A	3946 04	CMP DWORD PTR DS:[ESI+4],EAX
010FE03D	^75 E7	JNZ SHORT explorer.010FE026

위 그림이 바로 Name Pointer Table에서 GetProcAddress()함수의 포인터 주소를 찾기 위해서 문자열을 비교하는 과정입니다.



Address	Hex dump	Disassembly
010FE03F	8B5A 24	MOV EBX,DWORD PTR DS:[EDX+24]
010FE042	03DD	ADD EBX,EBP
010FE044	66:8B0C4B	MOV CX,WORD PTR DS:[EBX+ECX*2]
010FE048	8B5A 1C	MOV EBX,DWORD PTR DS:[EDX+1C]
010FE04B	03DD	ADD EBX,EBP
010FE04D	8B048B	MOV EAX,DWORD PTR DS:[EBX+ECX*4]
010FE050	03C5	ADD EAX,EBP

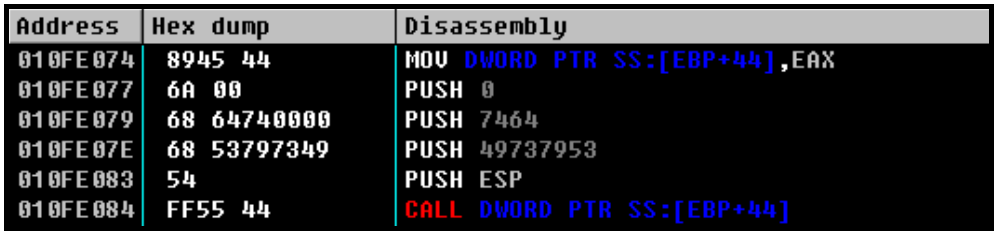
[EDX+24]에는 Kernel32.dll의 Oridinal Table의 RVA, [EBX+ECX*2]에는 GetProcAddress()함수의 Ordinal값인 0x0197h, [EDX+1C]에는 Address Table RVA가 저장됩니다. 마지막으로 [EBX+ECX*4]에는 GetProcAddress()함수의 주소포인터가 저장됩니다.



Address	Hex dump	Disassembly
010FE052	55	PUSH EBP
010FE053	81EC 90000000	SUB ESP,90
010FE059	8BEC	MOV EBP,ESP
010FE05B	8945 40	MOV DWORD PTR SS:[EBP+40],EAX
010FE05E	6A 00	PUSH 0
010FE060	68 61727941	PUSH 41797261
010FE065	68 4C696272	PUSH 7262694C
010FE06A	68 4C6F6164	PUSH 64616F4C
010FE06F	54	PUSH ESP
010FE070	57	PUSH EDI
010FE071	FF55 40	CALL DWORD PTR SS:[EBP+40]

결론은 [EBP+40]=EAX=7C80AC28 (kernel32.GetProcAddress)가 저장되고 GetProcAddress()함수를 사용하여 LoadLibrary()함수의 주소 포인터를, 즉 LoadLibrary()함수의 Entry Point를 얻어 옵니다. 그 전에 CALL DWORD PTR SS:[EBP+40]에는 앞서 설명한 것과 같이 GetProcAddress()함수의 Entry Pointer가 저장되는데 이 때 F7를 눌러 해당 함수로 진입해보면 아래와 같음을 알 수가 있습니다.

```
7C80AC28 > 8BFF          MOV EDI,EDI                ; kernel32.7C800000
7C80AC2A   55            PUSH EBP
7C80AC2B   8BEC          MOV EBP,ESP
7C80AC2D   51            PUSH ECX
7C80AC2E   51            PUSH ECX
7C80AC2F   53            PUSH EBX
```



Address	Hex dump	Disassembly
010FE074	8945 44	MOV DWORD PTR SS:[EBP+44],EAX
010FE077	6A 00	PUSH 0
010FE079	68 64740000	PUSH 7464
010FE07E	68 53797349	PUSH 49737953
010FE083	54	PUSH ESP
010FE084	FF55 44	CALL DWORD PTR SS:[EBP+44]

마찬가지로 [EBP+44]=EAX= 7C801D77(kernel32.LoadLibraryA)가 저장되고 CALL DWORD PTR SS:[EBP+44]는 LoadLibraryA()함수를 호출합니다. 이 때 F7를 눌러 해당 함수로 진입해보면 아래와 같음을 알 수가 있습니다. ESP=0007FEE4, (ASCII "SysIdt")

```
7C801D77 > 8BFF          MOV EDI,EDI                    ; kernel32.7C800000
7C801D79   55            PUSH EBP
7C801D7A   8BEC          MOV EBP,ESP
7C801D7C   837D 08 00    CMP DWORD PTR SS:[EBP+8],0
7C801D80   53            PUSH EBX
7C801D81   56            PUSH ESI
```

상기 루틴은 Win-Trojan/Agent.52736.FU가 생성한 %SYSTEM%₩Sysldt.dll를 LoadLibrary()함수를 사용하여 Explorer.exe가 로딩하는 과정입니다. 이후 Explorer.exe에 로딩된 DLL 모듈을 보면 아래와 같이 Sysldt.dll이 로딩되어 있음을 확인할 수 있습니다.

```
Executable modules, item 42
Base=7C9E0000
Size=00016000 (90112.)
Entry=7C9E7FD2 Sysldt.<ModuleEntryPoint>
Name=Sysldt   (system)
Path=C:₩WINDOWS₩system32₩Sysldt.dll
```

```
Address   Hex dump          Disassembly
010FE087  6A 00             PUSH 0
010FE089  68 496E6974       PUSH 74696E49
010FE08E  54                PUSH ESP
010FE08F  50                PUSH EAX
010FE090  FF55 40           CALL DWORD PTR SS:[EBP+40]
010FE093  FFD0              CALL EAX
010FE095  8BE5              MOV ESP,EBP
010FE097  81C4 90000000     ADD ESP,90
010FE09D  61                POPAD
010FE09E -E9 AB01F2FF       JMP explorer._ModuleEntry@0
```

74696E49(ASCII : Init)로 Sysldt.dll가 가지고 있는 Export 함수 중에 하나이고 PUSH EAX EAX=7C9E0000 (Sysldt.7C9E0000)는 메모리에 로딩된 Sysldt.dll의 Base Address가 되겠습니다. 즉 GetProcAddress()함수를 사용해서 Sysldt.dll의 Export함수 Init()의 Entry Pointer를 얻어 온 후 CALL EAX를 통해서 해당 함수를 실행하겠다는 의미가 되겠습니다.

```
010FE093   FFD0          CALL EAX                       ; Sysldt.Init
```

```
7C9E10F0 > E8 58080000       CALL Sysldt.7C9E194D
7C9E10F5   66:833D 34E59E7C>CMP WORD PTR DS:[7C9EE534],0
7C9E10FD   74 05             - JE SHORT Sysldt.7C9E1104
7C9E10FF   E8 35610000       CALL Sysldt.ResetSSDT
```

참고로 %SYSTEM%₩Sysldt.dll가 가지고 있는 Export 함수 테이블을 살펴 보면 아래와 같습니다. Init()함수 외에도 여러 Export 함수가 존재하는데 간단하게 분석을 해본 결과 InfectFile()란 Export 함수가 Explorer.exe에 바이러스 코드를 삽입하는 역할을 합니다.

pFile	Data	Description	Value
0000C1E8	00001078	Function RVA	0001 Entry
0000C1EC	00003A0C	Function RVA	0002 _EventLogon@4
0000C1F0	00003A0C	Function RVA	0003 _EventStartup@4
0000C1F4	000010EA	Function RVA	0004 GetCfg
0000C1F8	00003C9E	Function RVA	0005 InfectFile
0000C1FC	000010F0	Function RVA	0006 Init
0000C200	00004E22	Function RVA	0007 InstallHook
0000C204	00007239	Function RVA	0008 ResetSSDT
0000C208	00004E59	Function RVA	0009 UnHook
0000C20C	00007405	Function RVA	000A Uninstall

백신에서 %SYSTEM%₩Sysldt.dll를 진단 및 삭제했을 경우 Explorer.exe가 실행되지 않는 문제의 원인은 여기에 있습니다.
Patched.H는 Sysldt.dll의 Export 함수 Init를 실행한 후에 010FE09E -E9 AB01F2FF JMP explorer._ModuleEntry@0 주소로 점프(Explorer.exe의 OEP)하여 Explorer.exe가 정상으로 실행되도록 합니다.

그러나 백신에서 %SYSTEM%₩Sysldt.dll를 진단 및 삭제하게 되면 아래 루틴은 실패하므로 이후 Explorer.exe의 OEP로 점프하는 과정은 당연히 실패하므로 부팅 시에 Explorer.exe에 에러가 발생하면서 정상적으로 실행되지 않습니다.

>>>> %SYSTEM%₩Sysldt.dll로딩이 실패할 경우 <<<<

```
010FE077   6A 00          PUSH 0
010FE079   68 64740000    PUSH 7464
010FE07E   68 53797349    PUSH 49737953
010FE083   54             PUSH ESP
010FE084   FF55 44        CALL DWORD PTR SS:[EBP+44]

010FE087   6A 00          PUSH 0
010FE089   68 496E6974    PUSH 74696E49 ; 74696E49 = ASCII : Init
010FE08E   54             PUSH ESP
010FE08F   50             PUSH EAX  ; EAX=00000000, 성공했다면 00이 아닌 메모리에 로딩된 Sysldt.dll의 Base Address
010FE090   FF55 40        CALL DWORD PTR SS:[EBP+40]
010FE093   FFD0           CALL EAX ; EAX=00000000, 성공했다면 00이 아닌 Init함수의 Entry Point를 호출
```

Explorer.exe의 OEP(Original Entry Point) :

```
0101E24E >   8BFF         MOV EDI,EDI                    ;  kernel32.7C800000
0101E250   . 55           PUSH EBP
0101E251   . 8BEC         MOV EBP,ESP
0101E253   . 83EC 44      SUB ESP,44
0101E256   . 56           PUSH ESI
0101E257   . 57           PUSH EDI
```

%SYSTEM%₩Sysldt.dll가 진단 및 삭제된 후 재 부팅하면 아래 현상이 발생할 수 있습니다.