

Pin툴을 활용한 보안 USB 분석

임세진

https://youtu.be/h5K0skC_gQE

Contents

01. 분석 순서

02. Lexar USB 분석

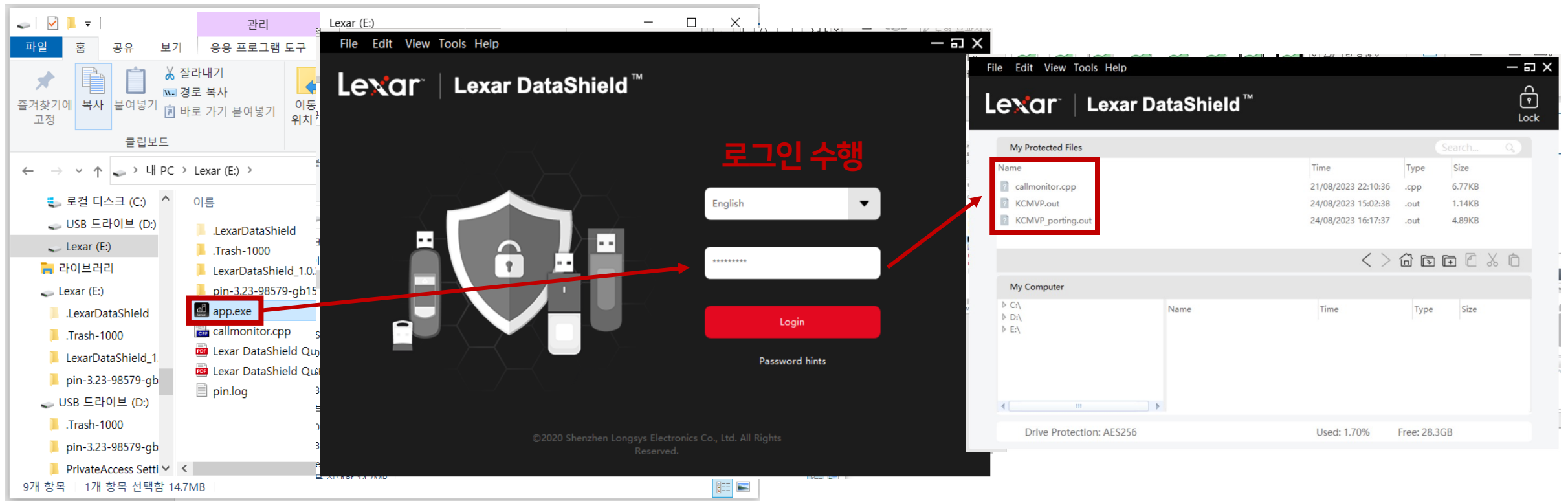
03. SanDisk USB 분석



01. 분석 순서

<USB의 보안 프로그램 동작 방식 - Lexar>

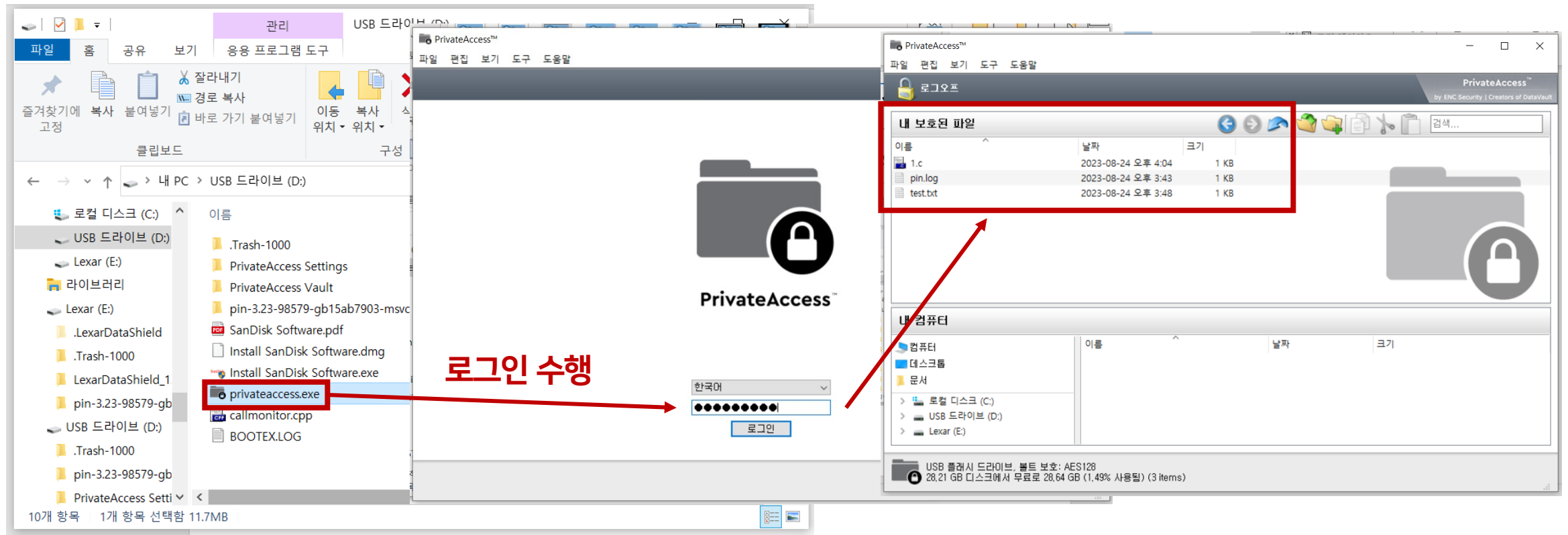
암호화하고자 하는 파일을 보안 프로그램에서 제공하는 드라이브에 저장



01. 분석 순서

<USB의 보안 프로그램 동작 방식 - SanDisk>

대부분의 보안 SW가 Windows, MacOS만 지원 → PIN 톨을 사용할 수 있는 Windows에서 분석 수행



01. 분석 순서

1. Target DLL이 주어지지 않았으므로 분석 대상으로 삼을 DLL을 식별해야 함

→ proccount.dll 활용

→ 사용된 DLL명과 함수명을 확인하여 Crypt, Hash, Key, Security 등 암호 관련 함수가 포함된 DLL 식별

2. 1에서 리스트업한 암호화 관련 DLL에 포함된 함수에 콜백 함수 등록 by 후킹

→ 함수명 옆에 DLL명도 같이 출력할 수 있도록 코드 수정

3. 실시간으로 프로그램의 암호화 기능을 수행하면서 분석

proccount.out - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

Procedure	Image	Address	Calls Instructions
DllUnregisterServer	C:\Windows\SysWOW64\IME\SHARED\WIMJKAPI.DLL	6e55c3b0	
unnamedImageEntryPoint	C:\Windows\SysWOW64\IME\SHARED\WIMJKAPI.DLL	6e54fd0	
DllCanUnloadNow	C:\Windows\SysWOW64\IME\SHARED\WIMJKAPI.DLL	6e54abf0	
DllGetClassObject	C:\Windows\SysWOW64\IME\SHARED\WIMJKAPI.DLL	6e548400	
.text	C:\Windows\SysWOW64\IME\SHARED\WIMJKAPI.DLL	6e544784	
.text	C:\Windows\SysWOW64\IME\SHARED\WIMJKAPI.DLL	6e5425d4	
unnamedImageEntryPoint	C:\Windows\SysWOW64\msvcvcp110_win.dll	734a6de0	
?_Init_dtor@Init@ios_base@std@@@CAXPAV123@@@Z	C:\Windows\SysWOW64\msvcvcp110_win.dll		
??1_UShinit@std@@@QAE@XZ	C:\Windows\SysWOW64\msvcvcp110_win.dll	734a5510	
??1_Lockit@std@@@QAE@XZ	C:\Windows\SysWOW64\msvcvcp110_win.dll	734a4d20	
??1_Init_locks@std@@@QAE@XZ	C:\Windows\SysWOW64\msvcvcp110_win.dll	734a4cf0	
??0_Lockit@std@@@QAE@H@Z	C:\Windows\SysWOW64\msvcvcp110_win.dll	734a4c90	
??0_Init_locks@std@@@QAE@XZ	C:\Windows\SysWOW64\msvcvcp110_win.dll	734a4c50	
?_Setgloballocale@locale@std@@@CAXPAX@Z	C:\Windows\SysWOW64\msvcvcp110_win.dll		

1번

```
VOID onImgLoad(IMG img, VOID* v){
    if (IMG_IsMainExecutable(img)) {
        target_start_addr = IMG_LowAddress(img);
        target_end_addr = IMG_HighAddress(img);
        TraceFile << "exe start: " << hex << target_start_addr << " end: " << target_end_addr << endl;
    }
    else if(IMG_Name(img).find(dll_0) != string::npos || IMG_Name(img).find(dll_1) != string::npos) {
        get_func_names(img);

        for(SEC sec = IMG_SecHead(img); SEC_Valid(sec); sec = SEC_Next(sec)) {
            for(RTN rtn = SEC_RtnHead(sec); RTN_Valid(rtn); rtn = RTN_Next(rtn)) {
                RTN_Open(rtn);
                set<string>::iterator it = fun_names.find(RTN_Name(rtn));

                if(it != fun_names.end()) {
                    RTN_COUNT* rc = new RTN_COUNT;
                    rc->_name = RTN_Name(rtn);
                    rc->_img = IMG_Name(img);
                    RTN_InsertCall(rtn, IPOINT_AFTER, (AFUNPTR)docount1, IARG_PTR, &(rc->_name), IARG_PTR, &(rc->_img), IARG_CONTEXT, IARG_END);
                }
                RTN_Close(rtn);
            }
        }
    }
}
```

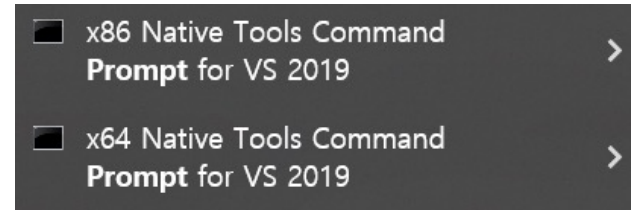
2번

```
VOID docount1(string* name, string* img, CONTEXT* ctx){
    TraceFile << *name << setw(60) << *img << endl;
}
```

01. 분석 순서

• Windows에서 PIN툴 사용 (리눅스보다 약간 까다로움)

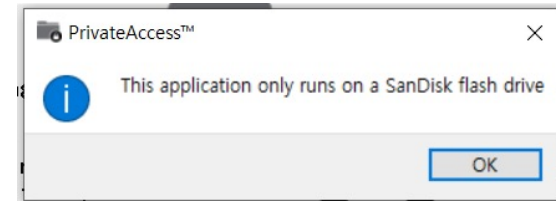
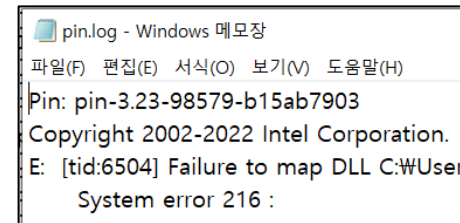
- make와 같은 컴파일 명령어를 사용하기 위해 Cygwin64 설치 및 환경변수 추가
- PIN 툴은 zip파일 설치 후 압축 해제하면 됨 (PIN 3.23 MSVC 설치함)
- PIN 툴을 컴파일하려면 visual studio command prompt를 사용해야 함



- x86 prompt를 사용하는 경우 make TARGET=ia32로 컴파일, x64 prompt는 TARGET=intel64로 컴파일

• 분석 대상 프로그램의 아키텍처에 따라 TARGET 선택

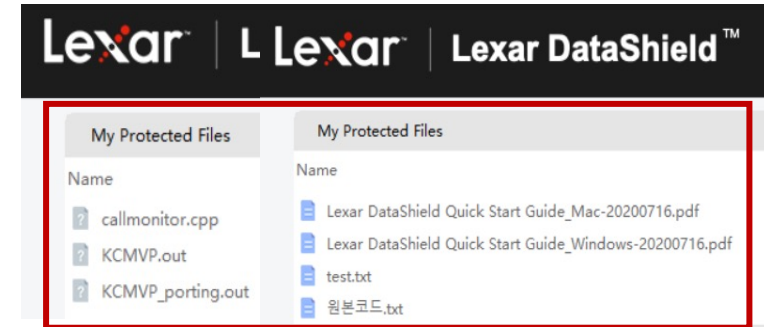
- 불일치 시 컴파일 에러 or 컴파일은 되지만 PIN 툴이 제대로 동작 X
- Lexar USB → x86 → TARGET=ia32
- SanDisk USB → x64 → TARGET=intel64



• SW를 실행시키는 작업 디렉터리가 해당 USB 드라이브여야 제대로 동작함

- PIN 툴 디렉터리를 USB 드라이브로 이동해서 실행시킴 (PIN 툴 절대 경로로 불러와도 됨)

C:\Users\wsejin>E:\app.exe E:\app.exe



02. Lexar USB

분석 대상으로 삼을 DLL :

```
string dll_0 = "bcryptprimitives.dll";  
string dll_1 = "advapi32.dll";
```

(1) 응용 프로그램이 할당된 메모리 영역으로 제한

- 아무 결과도 나오지 않음

(결론 및 분석)

- Lexar USB는 암호화 동작에 대한 구체적인 확인이 어려움 (암호화 수행 여부도 알 수 없었음)
- 난독화 가능성

```
unnamedImageEntryPoint C:\Users\sejin\Desktop\test\Project1\Win64\Debug\Lexar DataShield_windows.exe  
.text C:\Users\sejin\Desktop\test\Project1\Win64\Debug\Lexar DataShield_windows.exe  
.text C:\Users\sejin\Desktop\test\Project1\Win64\Debug\Lexar DataShield_windows.exe  
.text C:\Users\sejin\Desktop\test\Project1\Win64\Debug\Lexar DataShield_windows.exe
```

```
unnamedImageEntryPoint C:\Windows\SysWOW64\version.dll  
GetFileVersionInfoW C:\Windows\SysWOW64\version.dll  
unnamedImageEntryPoint C:\Windows\SysWOW64\oleacc.dll
```

IAT (Import Address Table)이 난독화 된 경우 or
PIN의 한계와 같은 이유로 unnamed~ 가 뜰 수 있다고 함

(2) 메모리 영역 제한 X

- 분석 대상 DLL에서 사용된 함수 중 보안 기능과 관련된 부분 (security, keyderivation)
- encrypt 관련 함수명은 찾을 수 없었음

StartServiceCtrlDispatcherA	C:\Windows\SysWOW64\advapi32.dll
ConvertStringSidToSidW	C:\Windows\SysWOW64\advapi32.dll
StartServiceCtrlDispatcherA	C:\Windows\SysWOW64\advapi32.dll
ConvertStringSecurityDescriptorToSecurityDescriptorW	C:\Windows\SysWOW64\advapi32.dll
ConvertStringSecurityDescriptorToSecurityDescriptorW	C:\Windows\SysWOW64\advapi32.dll
StartServiceCtrlDispatcherA	C:\Windows\SysWOW64\advapi32.dll
ConvertStringSecurityDescriptorToSecurityDescriptorW	C:\Windows\SysWOW64\advapi32.dll
ConvertStringSecurityDescriptorToSecurityDescriptorW	C:\Windows\SysWOW64\advapi32.dll
ConvertStringSecurityDescriptorToSecurityDescriptorW	C:\Windows\SysWOW64\advapi32.dll
ConvertSidToStringSidW	C:\Windows\SysWOW64\advapi32.dll
ProcessPrng	C:\Windows\SysWOW64\bcryptprimitives.dll
GetKeyDerivationInterface	C:\Windows\SysWOW64\bcryptprimitives.dll
ProcessPrng	C:\Windows\SysWOW64\bcryptprimitives.dll

03. SanDisk USB

분석 대상으로 삼을 DLL :

```
string dll_0 = "bcryptPrimitives.dll";
string dll_1 = "WINTRUST.dll";
string dll_2 = "CRYPTSP.dll";
string dll_3 = "CRYPT32.dll";
string dll_4 = "bcrypt.dll";
```

(1) 응용 프로그램이 할당된 메모리 영역으로 제한

- 3줄만 출력됨 (Encrypt X)

```
exe start: 7ff6e23d0000 end: 7ff6e3d88fff
----- Order of API Call List -----
----- Order of API Call List -----
----- Order of API Call List -----
----- Order of API Call List -----
----- Order of API Call List -----
CryptAcquireContextW      C:\WINDOWS\SYSTEM32\CRYPTSP.dll
CryptGenRandom           C:\WINDOWS\SYSTEM32\CRYPTSP.dll
CryptReleaseContext      C:\WINDOWS\SYSTEM32\CRYPTSP.dll
```

(결론 및 분석)

- Lexar << SanDisk USB에서 분석이 잘 수행됨
- 호출되는 함수의 순서를 통해 암호 모듈의 동작에 대한 대략적인 파악 가능
- 여러 DLL을 섞어서 암호화 기능을 구현한 것으로 보임

(2) 메모리 영역 제한 X

- 암호화 수행 및 암호화 관련 캐시를 해제하는 등 구체적인 동작 프로세스 확인 가능

GetAsymmetricEncryptionInterface	C:\WINDOWS\System32\bcryptPrimitives.dll
unnamedImageEntryPoint	C:\WINDOWS\System32\bcryptPrimitives.dll
unnamedImageEntryPoint	C:\WINDOWS\SYSTEM32\CRYPTSP.dll
unnamedImageEntryPoint	C:\WINDOWS\System32\WINTRUST.dll
unnamedImageEntryPoint	C:\WINDOWS\System32\WINTRUST.dll
unnamedImageEntryPoint	C:\WINDOWS\System32\WINTRUST.dll
unnamedImageEntryPoint	C:\WINDOWS\System32\WINTRUST.dll
WVTAsn1SpcPelmageDataEncode	C:\WINDOWS\System32\WINTRUST.dll
WVTAsn1SpcPelmageDataEncode	C:\WINDOWS\System32\WINTRUST.dll
WVTAsn1SpcPelmageDataEncode	C:\WINDOWS\System32\WINTRUST.dll
unnamedImageEntryPoint	C:\WINDOWS\System32\WINTRUST.dll
WVTAsn1SpcPelmageDataEncode	C:\WINDOWS\System32\WINTRUST.dll
unnamedImageEntryPoint	C:\WINDOWS\System32\WINTRUST.dll
BCryptVerifySignature	C:\WINDOWS\System32\bcrypt.dll
BCryptVerifySignature	C:\WINDOWS\System32\bcrypt.dll
unnamedImageEntryPoint	C:\WINDOWS\System32\bcrypt.dll
unnamedImageEntryPoint	C:\WINDOWS\System32\CRYPT32.dll
CryptFreeLruCache	C:\WINDOWS\System32\CRYPT32.dll
CryptSIPGetSignedDataMsg	C:\WINDOWS\System32\CRYPT32.dll

감사합니다