[보안 위협 분석 보고서]

윈도우 환경에서 애플리케이션의 DLL 하이재킹 취약점 상세 분석 보고서

2010. 08. 24

(주)하우리 사전대응팀

본 정보는 보안 위협에 대한 신속한 정보 공유를 위해 각 정부기관 및 기업 정보보호 담당자를 대상으로 배포됩니다.

목 차

1. 보안 위협 동향 분석

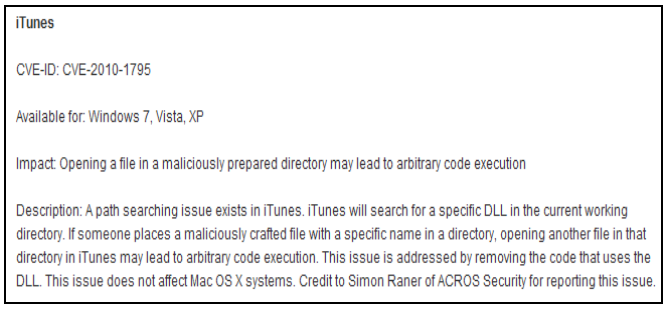
1-1. 윈도우 환경에서 애플리케이션의 DLL 하이재킹 취약점 이슈

8월 23일(미국 시간) MS는 윈도우 상의 특정 애플리케이션에서 DLL을 로딩할 때 적절한 경로 검증을 하지 않아 원격 코드의 실행이 가능한 취약점에 대한 보안 권고(2269637)를 발표했다. "DLL Preloading" 또는 "Binary Planting" 이라고도 불리는 DLL 하이재킹(Hijacking) 공격은 윈도우 운영체제에만 적용되거나 새롭게 발견된 공격 기법은 아니다. 이 공격 기법은 취약점이 존재하는 애플리케이션이 DLL 파일을 로딩할 때 악성 DLL 파일을 정상적인 DLL 파일로 인식시켜 로딩시키는 것이 핵심이다. MS에서는 해당 취약점을 가지고 있는 자사 제품들에 대한 추가적인 조사를 수행 중에 있다.

이와 관련하여 Rapid7의 보안 연구자인 HD Moore는 최근에 발표되었던 LNK 제로데이 취약점을 연구하는 과정에서 40개의 윈도우 애플리케이션에 원격 코드 실행 공격이 가능한 문제를 발견했다. HD Moore는 이와 같은 사실을 자신의 트위터를 통해 알렸다.

<HD Moore 트위터>

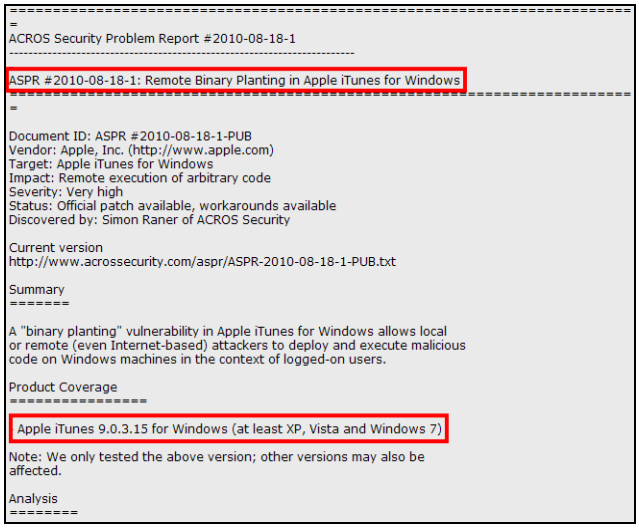
또한, 8월 12일 발표된 애플 아이튠즈 9.1의 보안 업데이트에서도 DLL 하이재킹과 관련된 내용을 찾아볼 수 있다.



<애플 아이튠즈 9.1 보안 업데이트>

해당 취약점은 아크로스 시큐리티(ACROS Security)가 보고하였으며 아이튠즈 이외에도 현재 다수의 윈도우 애플리케이션이 해당 취약점에 노출되어 있어 보안 문제가 우려되고 있다.

다음은 애플 아이튠즈의 DLL 하이재킹 취약점에 대한 아크로스 시큐리티의 보안 권고문이다.

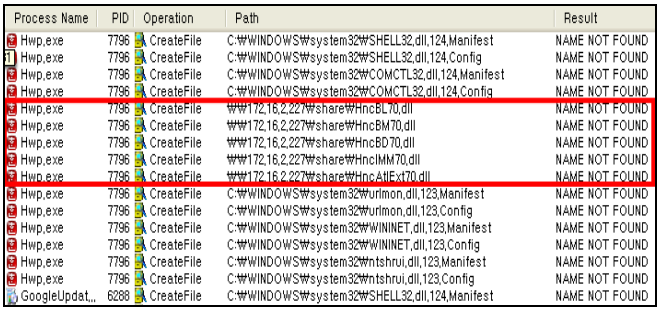


<아크로스 시큐리티의 애플 아이튠즈 취약점 보안 권고문>

공격자는 취약점이 존재하는 애플리케이션을 구동시킬 수 있는 문서, 이미지, 압축 파일 등과 같은 연결 파일과 악성 DLL을 WebDAV 또는 네트워크 공유 폴더에 심어 놓고, 사용자의 실행을 유도하여 악의적인 코드를 실행시킬 수 있다. 해당 취약점을 악용하면 짧은 시간에 많은 수의 PC가 감염될 수 있어 인터넷 웜으로 확대될 위험성도 존재한다.

윈도우 애플리케이션은 정규화된 경로를 지정하지 않은 상태에서 DLL 파일을 로딩하면 정의된 디렉토리를 대상으로 파일을 검색하여 DLL 파일을 로딩한다. 만약, 어떠한 디렉토리에서도 DLL 파일을 찾지 못하면 DLL 로딩 실패를 사용자에게 나타낸다.

다음 화면은 한글 문서 파일을 윈도우 공유 폴더에서 실행한 화면이다. 아래와 같이 애플리케이션이 사용하는 DLL 파일을 현재 작업 경로에서 호출한다.



<ProcessMonitor - DLL 파일 로드 모니터링>

공격자가 DLL 하이재킹 공격을 성공하기 위해서는 WebDAV나 SMB 공유 폴더에 접근이 가능해야 한다. 또한, 취약한 애플리케이션이 로드하는 DLL 이름과 동일한 파일 이름을 설정하여 악성 DLL 파일을 로딩시킬 수 있어야 한다.

※ 확대보기를 수행하면 이미지를 크게 볼 수 있습니다.

지난 2월과 7월 캘리포니아 다비스 대학의 권태호(Taeho Kwon) 박사는 이와 같은 내용을 국제 컨퍼런스를 통해서 발표했다. 권태호 박사의 논문은 ISSTA(International Symposium on Software Testing and Analysis)에 제출되었으며, MS의 MSRC에도 버그 리포트를 제출한 상태이다.

권태호 박사의 논문은 윈도우 애플리케이션이 DLL 파일을 로드할 때 발생하는 구조적인 문제점과 28개 애플리케이션에서 발견된 1700개의 버그가 어떻게 해커들에게 악용될 수 있는지를 자세히 설명하고 있다. 그의 연구 결과에 따르면 우리가 주로 사용하고 있는 MS Office 2007, 인터넷 익스플로러 8, 파이어폭스, 크롬, 오페라를 비롯하여 PDF 뷰어 프로그램인 어도비 리더, Foxit 리더, 메신저 프로그램 등 대부분의 윈도우 애플리케이션들이 해당 취약점에 노출된 것으로 확인되었다.

이번 취약점은 MS의 보안 패치만으로는 완벽하게 해결되지 않을 것으로 예상되며, 취약점이 존재하는 애플리케이션 업체에서도 별도의 패치가 필요할 것으로 예상된다. 이와 같은 공격으로 인한 피해를 최소화하기 위해서는 방화벽에서 SMB, WebDAV의 아웃바운드 트래픽을 차단하여 악성 DLL 파일이 실행되는 것을 예방하고, PC에 기본적으로 설정되어 있는 "Web Client" 서비스를 정지시켜야 한다.

■ **DLL 하이재킹 공격에 대한 사용자 임시대응 방안**

MS에서 공식 패치를 제공하기 전까지 WebDAV와 원격 네트워크 공유로부터 라이브러리 로딩을 비활성화하기 위해, DLL 검색 경로 알고리즘을 제어하는 CWDIllegalInDllSearch 레지스트리 키를 생성하여 임시로 조치할 수 있다.

CWDIllegalInDllSearch 레지스트리 키는 다음 경로에 추가될 수 있다.

모든 디렉토리에 대해 레지스트리 키를 적용하려면 다음과 같은 경로에 설정한다.

HKEY_LOCAL_MACHINE₩SYSTEM₩CurrentControlSet₩Control₩Session Manager

지정된 응용 프로그램에 대해서 레지스트리 키를 적용하려면 다음과 같은 경로에 설정한다.

HKEY_LOCAL_MACHINE₩Software₩Microsoft₩Windows NT₩CurrentVersion₩Image File Execution Options₩<애플리케이션 바이너리 이름>

CWDIllegalInDllSearch 레지스트리 키 값은 LoadLibrary 및 LoadLibraryEx의 동작을 다음과 같이 정의한다.

1. 응용 프로그램이 C:₩Program Files와 같은 로컬 폴더에서 시작되는 경우

CWDIllegalInDllSearch 값	LoadLibrary 및 LoadLibraryEx에서 DLL 검색 경로의 동작
0xFFFFFFFF	기본 DLL 검색 순서에서 현재 작업 디렉토리 제거
0	기본 DLL 검색 경로 사용
1	현재 작업 디렉토리가 WebDAV 폴더로 설정된 경우 현재 작업 디렉토리에서의 DLL 로드 차단
2	현재 작업 디렉토리가 원격 폴더로 설정된 경우 현재 작업 디렉토리에서의 DLL 로드 차단
키 또는 다른 값 없음	기본 DLL 검색 경로 사용

2. 응용 프로그램이 ₩₩remote₩shareremote₩share와 같은 원격 폴더에서 시작되는 경우

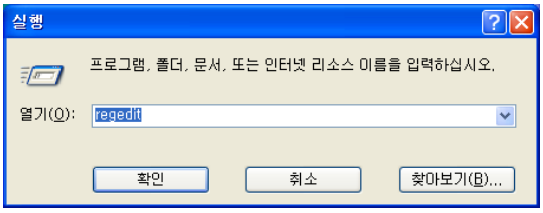
CWDIllegalInDllSearch 값	LoadLibrary 및 LoadLibraryEx에서 DLL 검색 경로의 동작
0xFFFFFFFF	기본 DLL 검색 순서에서 현재 작업 디렉토리 제거
0	기본 DLL 검색 경로 사용
1	현재 작업 디렉토리가 WebDAV 폴더로 설정된 경우 현재 작업 디렉토리에서의 DLL 로드 차단
2	현재 작업 디렉토리가 원격 폴더로 설정된 경우 현재 작업 디렉토리에서 DLL을 로드할 수 있도록 허용합니다. WebDAV 공유에서 로드된 DLL은 CWD가 WebDAV 공유로 설정된 경우 차단됩니다.
키 또는 다른 값 없음	기본 DLL 검색 경로 사용

3. 응용 프로그램이 http://remote/share와 같은 WebDAV 폴더에서 시작되는 경우

CWDIllegalInDllSearch 값	LoadLibrary 및 LoadLibraryEx에서 DLL 검색 경로의 동작
0xFFFFFFFF	기본 DLL 검색 순서에서 현재 작업 디렉토리 제거
키 또는 다른 값 없음	기본 DLL 검색 경로 사용

➢ **모든 응용 프로그램이 WebDAV 공유에서 DLL을 로드할 수 없게 설정하는 방법**

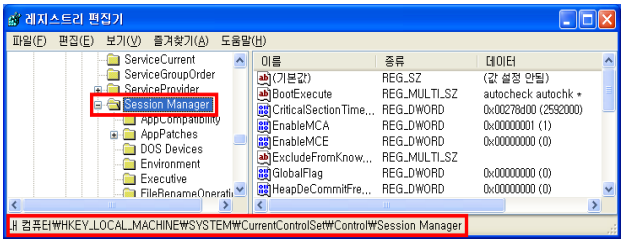
1. 시작-> 실행에 regedit를 입력하여 레지스트리 편집기를 실행한다.



<레지스트리 편집기 실행>

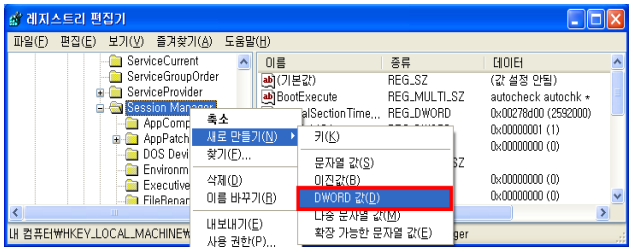
2. 레지스트리 편집기를 열어 다음의 레지스트리 키 경로로 이동한다.

HKEY_LOCAL_MACHINE₩SYSTEM₩CurrentControlSet₩Control₩Session Manager



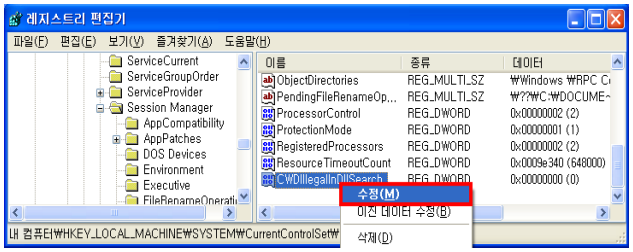
<레지스트리 경로 이동>

3. Session Manager를 마우스 오른쪽 단추로 클릭하고 새로 만들기를 가리킨 다음 DWORD 값을 클릭한다.



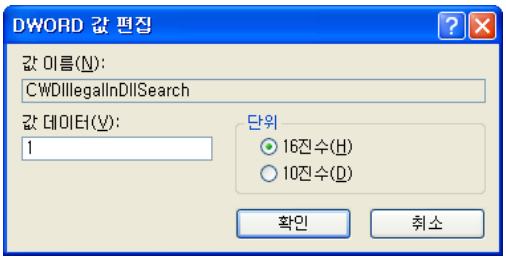
<레지스트리 키 생성>

4. CWDIllegalInDllSearch를 입력한 후 수정을 클릭한다.



<레지스트리 키 수정>

5. DWORD 값 편집 창에서 1을 입력한 다음 확인을 클릭한다.



<레지스트리 키 DWORD 값 설정>

특정 응용 프로그램의 WebDAV 공유에서 DLL을 로드할 수 없게 설정하거나, 원격(SMB) 공유에서 DLL을 로드할 수 없게 설정하기 위한 자세한 내용은 Microsoft Knowledge Base Article 2264107 문서를 참고하기 바란다.

참고 : http://support.microsoft.com/kb/2264107

※ 확대보기를 수행하면 이미지를 크게 볼 수 있습니다.

2. 취약점 상세 분석

2-1. [MSA2269637] 안전하지 않은 라이브러리 로딩 취약점으로 인한 원격 코드 실행 문제점

(1) 취약점 개요

Windows 운영체제가 프로그램이 요청한 DLL 파일을 로딩하는 과정에서 발생하는 구조적 문제점으로 인해, 원격의 공격자로부터 악의적인 명령코드의 실행을 허용할 수 있는 문제점이 존재한다.

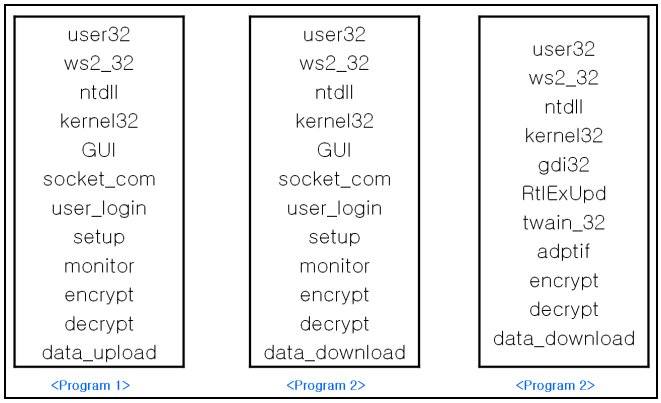
공격자는 WebDAV나 네트워크 공유 서비스를 연 후, 사용자로 하여금 DLL Preloading 취약점이 존재하는 프로그램의 실행을 유도하는 방식으로 공격을 수행한다.

현재 해당 취약점에 대한 패치가 발표되지 않았으며, Windows 운영체제에서 동작하는 프로그램 중 상당수가 해당 공격에 노출된 것으로 판단되고 있어 사용자의 주의가 필요하다.

취약점 기반 기술

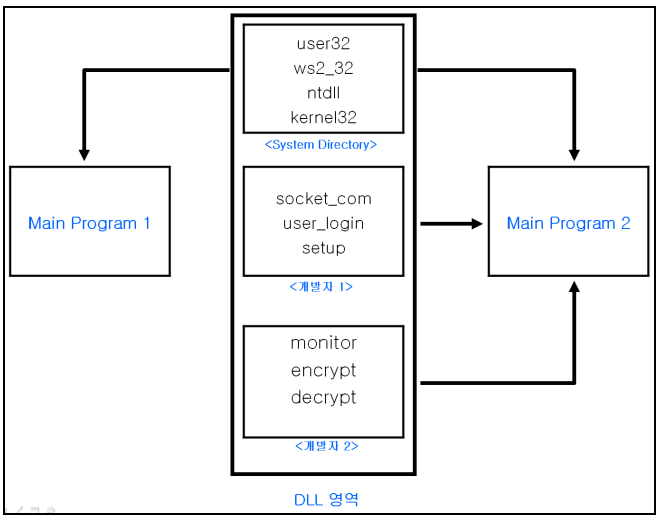
현대의 프로그램들은 실행파일 하나로 간단하게 이루어지는 경우는 거의 없다. 프로그램들은 DLL 이라는 파일을 통해 모듈화되어, 여러 개의 파일로 나누어져 구성된다.

DLL을 사용하지 않을 경우, 프로그램이 사용하기 위한 모든 함수를 파일 안에 포함해야 하기 때문에 파일사이즈가 비정상적으로 커지는 것은 물론, 중복된 코드가 반복적으로 사용되어 효율성이 떨어지는 문제가 발생한다.



<DLL을 사용하지 않을 경우 프로그램의 구조>

Windows 운영체제는 DLL 이라는 기능을 지원함으로써 개발에 필요한 필수적인 API 함수를 공유할 수 있도록 하여 프로그램의 크기를 획기적으로 줄이고, 개발자 간의 작업 분담으로 프로그램을 더욱 더 정교하게 개발할 수 있도록 지원한다.



<DLL 서비스 개요>

DLL은 프로그램 개발 및 실행에 필수적인 요소로 인식되고 있으며, 실제 프로그램을 실행한 후 확인해보면 프로그램의 크기에 따라 적게는 3~4개에서 많게는 수백 개의 DLL을 호출한다.

프로그램은 LoadLibrary API 함수를 사용해 DLL 파일을 프로그램의 메모리로 로드할 수 있다. 해당 함수의 사용법은 아래와 같다.

```
HMODULE handle = LoadLibrary("test.dll");
```

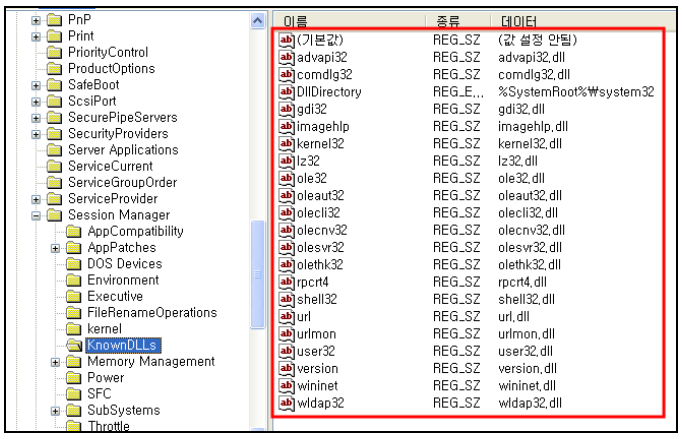
<LoadLibrary API 사용 예제>

LoadLibrary 함수의 인자에 경로정보 없이 파일명만 입력되어 있는 것을 확인할 수 있다. 원칙적으로 파일의 전체 경로가 입력되어야 하지만, 경로가 존재하지 않을 경우 자동으로 파일이 탐색되기 때문에 대부분의 코딩이 위와 같은 형태로 이루어진다.

파일의 경로가 존재하지 않을 경우, 가장 먼저 레지스트리의 KnownDLLs 키가 조사된다. 해당 레지스트리의 경로 및 내용은 아래와 같다.

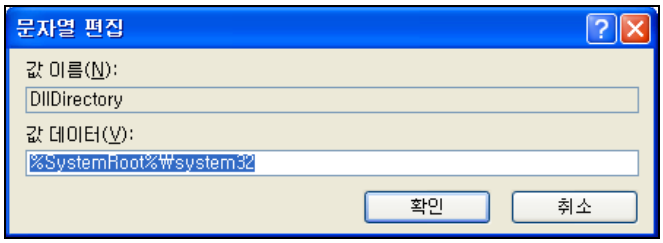
```
HKLM/System/CurrentControlSet/Control/Session Manager/KnownDLLs
```


※ 확대보기를 수행하면 이미지를 크게 볼 수 있습니다.



<KnownDLLs 목록>

레지스트리가 등록되어 있지 않으면 DllDirectory에 저장된 디렉토리를 검색한다.



<DllDirectory 값 확인>

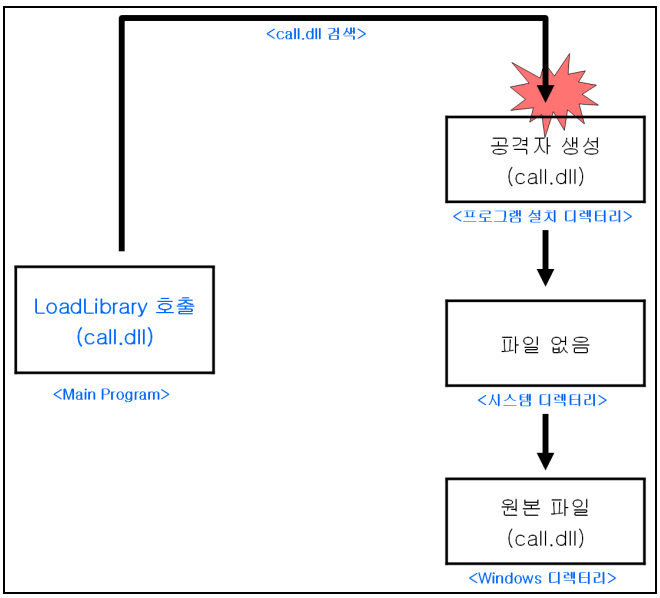
만약 로드하고자 하는 DLL이 해당 레지스트리에 존재하지 않으면, 다음과 같은 우선순위에 의해 디렉토리들이 검색되고 DLL이 동적으로 로드된다.

1. 프로그램이 설치된 디렉토리
2. GetSystemDirectory 함수를 통해 반환되는 시스템 디렉토리.
3. 16-bit 시스템 디렉토리
4. GetWindowsDirectory 함수를 통해 반환되는 윈도우즈 디렉토리
5. 현재 디렉토리
6. PATH 환경변수에 등록된 디렉토리

<DLL 호출 우선 순위>

(2) 취약점 상세 분석

공격자는 프로그램을 실행하여 실행되는 DLL 파일의 경로를 추적한 후, DLL의 우선순위가 낮을 경우 우선순위가 높은 디렉토리에 동일한 이름으로 악의적인 명령코드를 수행하는 DLL 파일을 생성한다.



<공격 순서도>

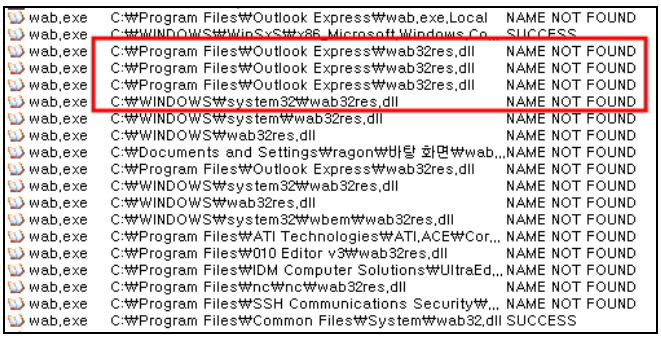
하지만 공격자가 권한이 없는 상태에서 사용자의 컴퓨터에 파일을 생성하는 것은 매우 힘들기 때문에, 대부분의 공격이 WebDAV나 네트워크 공유 등을 통한 원격에서 이루어진다.

WebDAV나 네트워크 공유를 통해 접속한 디렉토리는 DLL 호출 우선 순위의 5번 "현재 디렉토리"에 해당하기 때문에, 해당 공격이 성공하기 위해서는 호출되는 DLL이 우선순위가 가장 낮은 PATH 환경변수에 등록되어 있거나 존재하지 않는 DLL이어야 한다.

DLL 호출이 요청되었으나 존재하지 않는 경우, 6가지의 우선순위와 상관없이 생성과 동시에 코드를 실행할 수 있어 주요 분석 대상이 된다.

프로그램에서 호출되는 DLL이 수백 개에 이를 경우, 이를 제대로 제어하기가 힘들기 때문에 존재하지 않는 DLL을 호출할 가능성이 매우 높다. 만약 존재하지 않은 DLL에 대한 호출이 이루어질 경우, 해당 취약점을 통해 원격의 명령코드를 실행할 수 있다.

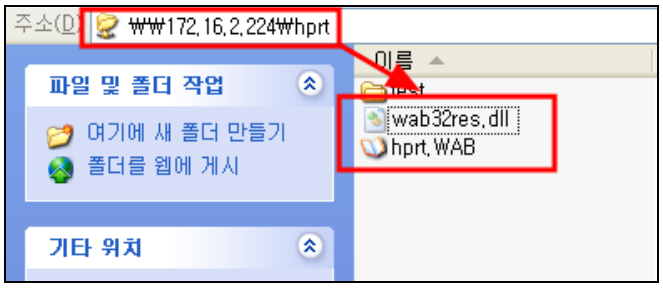
Process Monitor를 활용하면 존재하지 않는 DLL에 대한 호출을 쉽게 탐지할 수 있다.



<존재하지 않는 DLL 호출 탐지>

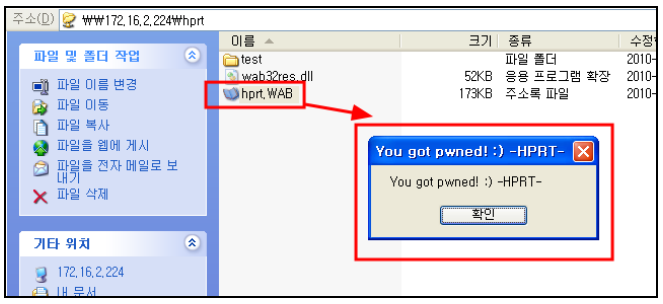
※ 확대보기를 수행하면 이미지를 크게 볼 수 있습니다.

Windows 운영체제에서 기본으로 제공하는 주소록 프로그램을 실행하면 wab32res.dll 파일의 경로를 찾을 수 없다는 로그를 확인할 수 있다.
해당 정보를 통해 네트워크 공유 서버를 구성한 후, 주소록 파일과 악의적인 명령코드에 해당하는 wab32res.dll 파일을 구성한다.



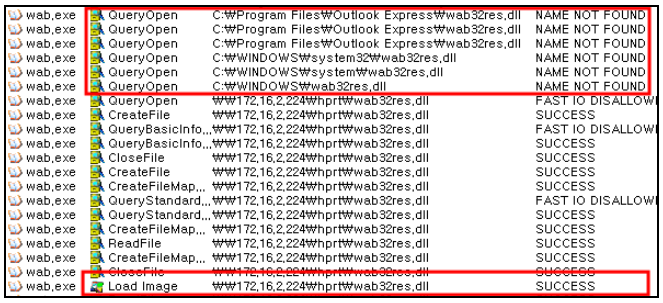
<공격환경 구현>

일반 사용자가 네트워크 공유 또는 WebDAV를 통해 공격자의 서버에 접속한 후, 주소록 파일을 실행할 경우 악의적인 dll 파일이 실행된다.



<명령코드 실행 확인>

악성코드가 실행되는 순간 Process Monitor를 통해, 공유폴더에 위치한 공격자의 dll이 실행되는 것을 확인할 수 있다.



<dll 로드 확인>

(3) 프로그램 개발 시 고려사항

(사용자의 대응방안은 동향 분석 정보 부분의 임시 대응방안을 참고한다.)

해당 취약점은 Windows 운영체제의 구조적인 문제점을 이용하는 것으로, 개발 단계에서 보안이 고려될 경우 해당 취약점을 해결할 수 있다.

1. 함수 사용 시 전체 경로 사용

LoadLibrary, CreateProcess 및 ShellExecute 등 다른 파일이나 프로세스를 메모리에 로드하는 함수를 사용할 때, 전체경로를 입력한다. 전체 경로가 입력 될 경우 해당 파일에 대한 직접적인 접근이 이루어지며, 다른 디렉토리에 대한 검색 기능은 동작하지 않는다.

전체 경로를 사용해 함수를 호출하는 예제코드는 다음과 같다.

```
HMODULE handle =
LoadLibrary("c:₩₩windows₩₩system32₩₩test.dll");
```
<전체 경로 사용 예제코드>

2. SetDllDirectory 함수 사용

SetDllDirectory 함수를 인자 없이 호출할 경우, 현재 파일이 실행되는 디렉토리를 검색경로에서 제외할 수 있다. 하지만 이 함수는 현재 프로세스뿐 아니라, 시스템에 동작하고 있는 모든 프로세스 및 스레드에 영향을 미치므로 함수 적용 전 시스템에 미칠 영향을 면밀하게 검토해야 한다.

해당 함수가 적용될 경우 공격자는 USB 및 원격 네트워크를 이용한 공격이 불가능 하며, 시스템 권한을 획득하여 프로그램이 설치된 디렉토리 및 시스템 디렉토리에 파일을 직접 설치해야 하므로 공격의 위험도를 상당부분 상쇄할 수 있다.

SetDllDirectory 함수를 사용해 디렉토리 검색 경로에서 현재 실행되는 파일의 경로를 제거하는 예제코드는 다음과 같다.

```
SetDllDirectory ("");
HMODULE handle = LoadLibrary("test.dll");
```
<SetDllDirectory 함수 사용 예제코드>

※ 확대보기를 수행하면 이미지를 크게 볼 수 있습니다.

3. 하우리 사전대응팀

하우리 사전대응팀은 취약점 등 최신 보안 위협의 분석을 통해 해킹 및 악성코드로 발전할 수 있는 가능성을 연구하여 사전에 예방하는 활동을 수행합니다.