(Process) Code Injection

Windows Platform

Code Injection

- Requirements for Injected Code
  - Must be self-loadable
  - Must be position independent
  - Must resolve dependencies by itself

Code Injection on Windows

- Open Process to Inject Code
  - OpenProcess(..)
- Allocate Memory on Remote Process
  - VirtualAllocEx(..)
- Write code on Remotely allocated memory
  - WriteProcessMemory(..)
- Start execution of injected code
  - CreateRemoteThread(..)

OpenProcess function

Opens an existing local process object.

Syntax

C++

```
HANDLE WINAPI OpenProcess(
  _In_  DWORD dwDesiredAccess,
  _In_  BOOL bInheritHandle,
  _In_  DWORD dwProcessId
);
```

VirtualAllocEx function

Reserves or commits a region of memory within the virtual address space of a specified process. The function initializes the memory it allocates to zero, unless **MEM_RESET** is used.

To specify the NUMA node for the physical memory, see **VirtualAllocExNuma**.

Syntax

C++

```
LPVOID WINAPI VirtualAllocEx(
  _In_      HANDLE hProcess,
  _In_opt_  LPVOID lpAddress,
  _In_      SIZE_T dwSize,
  _In_      DWORD flAllocationType,
  _In_      DWORD flProtect
);
```

WriteProcessMemory function

Writes data to an area of memory in a specified process. The entire area to be written to must be accessible or the operation fails.

Syntax

C++

```
BOOL WINAPI WriteProcessMemory(
  _In_   HANDLE hProcess,
  _In_   LPVOID lpBaseAddress,
  _In_   LPCVOID lpBuffer,
  _In_   SIZE_T nSize,
  _Out_  SIZE_T *lpNumberOfBytesWritten
);
```

CreateRemoteThread function

Creates a thread that runs in the virtual address space of another process.

Use the **CreateRemoteThreadEx** function to create a thread that runs in the virtual address space of another processor and optionally specify extended attributes.

Syntax

C++

```
HANDLE WINAPI CreateRemoteThread(
  _In_   HANDLE hProcess,
  _In_   LPSECURITY_ATTRIBUTES lpThreadAttributes,
  _In_   SIZE_T dwStackSize,
  _In_   LPTHREAD_START_ROUTINE lpStartAddress,
  _In_   LPVOID lpParameter,
  _In_   DWORD dwCreationFlags,
  _Out_  LPDWORD lpThreadId
);
```