Hiding program using process creation flags

Hiding program using dwCreationFlags :

```
BOOL CreateProcess(
 lpApplicationName,        // Path to the executable
lpCommandLine,             // Command-line arguments
lpProcessAttributes,     // Security attributes for the process
lpThreadAttributes,      // Security attributes for the primary thread
bInheritHandles,           // Inherit handles from parent process
dwCreationFlags,           // Process creation flags
lpEnvironment,             // Pointer to environment block
lpCurrentDirectory,        // Working directory of the new process
lpStartupInfo,         // Pointer to STARTUPINFO structure
lpProcessInformation     // Pointer to PROCESS_INFORMATION structure
);
```

Flag	Description
CREATE_NEW_CONSOLE	Creates a new console window for the child process.
CREATE_SUSPENDED	Creates the process in a suspended state. (Used in process injection.)
CREATE_NO_WINDOW	Runs the process without showing any window (used for stealth malware).
CREATE_DEFAULT_ERROR_MODE	Uses the parent's error mode.
CREATE_BREAKAWAY_FROM_JOB	Allows the process to run outside of a job object.
CREATE_SEPARATE_WOW_VDM	Runs the process in a separate virtual DOS machine (for 16-bit apps).
CREATE_SHARED_WOW_VDM	Runs in a shared DOS environment (legacy stuff).
DEBUG_PROCESS	Allows the parent process to debug the child process.
DEBUG_ONLY_THIS_PROCESS	Only allows the parent to debug this process.
DETACHED_PROCESS	Runs the process without associating with the parent console.
EXTENDED_STARTUPINFO_PRESENT	Allows passing extended startup information.
INHERIT_PARENT_AFFIINITY	Inherits the CPU affinity of the parent process.
CREATE_PROTECTED_PROCESS	Creates a protected process (used in anti-debugging).

Code :

```
#include <windows.h>
#include <stdio.h>

int main() {
    STARTUPINFO si = { sizeof(si) }; // setting zero
    PROCESS_INFORMATION pi = {0};    // setting zero


    // Create Notepad process
    if (CreateProcess(
        "C:\\Windows\\System32\\notepad.exe", // Application name
        NULL, // No command-line arguments
        NULL, NULL, FALSE, // Default security
        CREATE_SUSPENDED, // create in suspended mode
        NULL, NULL, // Use parent's environment and directory
        &si, &pi // Pass structures
    ))
{
        printf("[+] Process created successfully in suspended mode!\n");
        printf("[+] Process ID: %lu\n", pi.dwProcessId);
        printf("[+] Thread ID: %lu\n", pi.dwThreadId);

        // Close process handles
        CloseHandle(pi.hProcess);
        CloseHandle(pi.hThread);
    }
else {
        printf("[-] Failed to create process. Error: %lu\n", GetLastError());
    }
    return 0;
}
```

Code :

```
#include <windows.h>
#include <stdio.h>

int main() {
    STARTUPINFO si = { sizeof(si) }; // setting zero
    PROCESS_INFORMATION pi = {0};    // setting zero


    // Create Notepad process
    if (CreateProcess(
        "C:\\Windows\\System32\\cmd.exe", // Application name
        NULL, // No command-line arguments
        NULL, NULL, FALSE, // Default security
        CREATE_NO_WINDOW, // applicable for console app only
        NULL, NULL, // Use parent's environment and directory
        &si, &pi // Pass structures
    ))
{
        printf("[+] Process created successfully in suspended mode!\n");
        printf("[+] Process ID: %lu\n", pi.dwProcessId);
        printf("[+] Thread ID: %lu\n", pi.dwThreadId);

        // Close process handles
        CloseHandle(pi.hProcess);
        CloseHandle(pi.hThread);
    }
else {
        printf("[-] Failed to create process. Error: %lu\n", GetLastError());
    }
    return 0;
}
```