黑客常用 WINAPI 函数整理

一、进程

创建进程：

CreateProcess("C:\\windows\\notepad.exe",0,0,0,0,0,0,0,&si,&pi);

WinExec("notepad",SW_SHOW);

ShellExecute(0,"open","notepad","c:\\a.txt","",SW_SHOW);

ShellExecuteEx(&sei);

遍历进程：

CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS,0);

Process32First(hsnap,&pe32);

Process32Next(hsnap,&pe32);

终止进程：

ExitProcess(0);

TerminateProcess(hProc,0);

打开进程：

OpenProcess(PROCESS_ALL_ACCESS,0,pid);\

获取进程 ID：

GetCurrentProcesssId();

获取进程可执行文件路径：

GetModuleFileName(NULL,buf,len);

GetProcessImageFileName(hproc,buf,len);

遍历进程模块信息：

CreateToolhelp32Snapshot(TH32CS_SNAPMODILE,pid);

Module32First(hsnap,&mdl32);

Module32Next(hsnap,&mdl2);

获取指定模块句柄：

GetModuleHandle(“kernel32.dll”);

获取模块内函数地址：

GetProcessAddr(hmdl,”MessageBox”);

动态加载 DLL：

LoadLibrary(“user32.dll”);

卸载 DLL：

FreeLibrary(hDll);

获取进程命令行参数：

GetCommandLine();
任何进程 GetCommandLine 函数地址后偏移一个字节后的 4 字节地址为命令行地址。

读写远程进程数据：

ReadProcessMemory(hproc,baseAddr,buf,len,&size);

WriteProcessMemory(hproc,baseAddr,buf,len,&size);

申请内存：

VirtualAlloc(0,size,MEM_COMMIT, PAGE_EXECUTE_READWRITE);

VirtualAllocEx(hproc,0,size,MEM_COMMIT,
PAGE_EXECUTE_READWRITE);

修改内存属性：

VirtualProtect(addr,size,PAGE_EXECUTE_READWRITE,&oldAddr);

VirtualProtectEx(hproc,addr,size,PAGE_EXECUTE_READWRITE,&oldAd
dr);

释放内存：

VirtualFree(addr, size, MEM_RELEASE);

VirtualFreeEx(hproc, addr, size, MEM_RELEASE);

获取系统版本(Win NT/2K/XP<0x80000000)：

getVersion();

读写进程优先级：

SetPriorityClass(hproc,Normal);

GetPriority(hproc);

SetProcessPriorityBoost(hproc,true);

GetProcessPriorityBoost(hproc,pBool);

二、线程

创建线程(CreateThread 的线程函数调用了 strtok、rand 等需使用_endthread()释放内存)：

CreateThread(0,0,startAddr,&para,0,&tid);

_beginthread(startAddr,0,0);

_beginthreadex(0,0,startAddr,0,0,&tid);

CreateRemoteThread(hproc,0,0,func,&para,0,&tid);

获取线程 ID：

GetCurrentThreadId();

关闭线程句柄（减少内核对象使用次数，防止内存泄漏）：

CloseHandle(hthread);

挂起与激活线程(维护暂停次数)：

SuspendThread(hthread);

ResumeThread(hthread);

获取线程退出代码：

GetExitCode(hthread,&code);

等待线程退出(线程受信状态或超时)：

WaitForSignleObject(htread,1000);

WaitForMultipleObjects(num,handles,true,INFINITE);

遍历线程：

CreateToolhelp32Snapshot(TH32CS_SNAPTHREAD,0);

Thread32First(hsnap,&mdl32);

Thread32Next(hsnap,&mdl2);

获取线程函数入口：

ZwQueryInfomationThread(hthread,ThreadQuerySetWin32StartAddress,&buf,4,NULL);

打开线程

OpenThread(THREAD_ALL_ACCESS,false,&tid);

获取线程函数地址所属模块：

GetMappedFileName(hproc,addr,buf,256);

读写线程优先级：

SetThreadPriority(hthread,Normal);

GetThreadPriority(hthread);

SetThreadPriorityBoost(hproc,true);

GetThreadPriorityBoost(hproc,pBool);

终止线程：

ExitThread(5);

TerminateThread(hthread,5);

线程同步临界区对象：

InitializeCriticalSection(&cs);

EnterCriticalSection(&cs);

LeaveCriticalSection(&cs);

DeleteCriticalSection(&cs);

线程同步事件内核对象：

OpenEvent(EVENT_ALL_ACCESS,false,name);

CreateEvent(NULL,false,true,NULL);

WaitForSingleObject(hevnt,INFINITE);

SetEvent(hevnt);

ResetEvent(hevnt);

线程同步互斥内核对象：

CreateMutex(NULL,false,NULL);

WaitForSingleObject(hmutex,INFINITE);

ReleaseMutex(hmutex);

OpenMutex(MUTEX_ALL_ACCESS,false,name);

三、注册表

创建键：

RegCreateKeyEx(HKEY_CURRENT_USER,”TestNewKey”,0,0,REG_OPTION_VOLATILE,KEY_ALL_ACCESS,0,&subkey,&state);

打开键：

RegCreateKeyEx(HKEY_CURRENT_USER,”Control Panel”,0,KEY_ALL_ACCESS,&subkey);

关闭键：

RegCloseKey(hkey);

遍历键：

RegEnumKeyEx(hsubkey,index,keyname,&nameSize,0,0,0,&time);

FileTimeToSystemTime(&time,&systime);

RegQueryInfo(hsubkey,0,0,0,&count,0,0,0,0,0,0,0);

删除键：

RegDeleteKeyEx(hmainkey,subkeyName);

创建值：

RegSetValueEx(hsubkey,”test”,0,REG_WORD,(BYTE*)&value,4);

遍历值：

RegEnumValue(hsubkey,index,name,&nameSize,0,&type,valuebuf,valueLen);

RegQueryValueEx(hsubkey,name,0,type,buf,&size);

删除值：

RegDeleteValue(hsubkey,valuename);

四、文件

创建/打开文件：

CreateFile(“a.txt”,GENERIC_READ|GENERIC_WRITE,FILE_SHARE_READ,0,OPEN_EXISTING,FILE_ATTRIBUTE_NORMAL,0);

设置文件指针：

SetFilePointer(hFile,0,NULL,FILE_END);

读写文件：

ReadFile(hFile,buf,len,&size,0);

WriteFile(hFile,buf,len,&size,0);

强制文件写入磁盘，清空文件高速缓冲区：

FlushFileuffers(hFile);

[解]锁文件区域：

LockFile(hFile,0,0,100,0);

UnlockFile(hFile,0,0,100,0);

复制文件：

CopyFile(src,des,true);

CopyFileEx(src,des,func,&para,false, COPY_FILE_FAIL_IF_EXISTS);

移动文件：

MoveFile(src,des);

MoveFileEx(src,des,false);

MoveFileWithProgress(src,des,fun,&para,
MOVEFILE_COPY_ALLOWED);

删除文件：

DeleteFile(filename);

获取文件类型(FILE_TYPE_PIPE)：

GetFileType(hFile);

获取文件大小：

GetFileSize(hFile,&high);

获取文件属性(例如 FILE_ATTRIBUTE_DIRECTORY 进行&运算)：

GetFileAttributes(hFile);

遍历文件：

FindFirstFile(nameMode,&wfd);

FindNextFile(hFile,&wfd);

创建管道：

CreatePipe(&hRead,&hWrite,&sa,0);

创建内存映射文件：

CreateFile(“d:\\a.txt”,GENERIC_READ|GENERIC_WRITE,FILE_SHARE
_READ,0,OPEN_EXISTI
NG,FILE_ATTRIBUTE_NORMAL,”myMap”);

加载内存映射文件：

MapViewOfFile(hmap,FILE_MAP_ALL_ACCESS,0,0,0);

打开内存映射文件：

OpenFileMapping(FILE_AMP_ALL_ACCESS,false,”myMap”);

卸载内存映射文件：

UnmapViewOfFile(baseAddr);

强制写入内存映射文件到磁盘：

FlushViewOfFile(baseAddr,len);

创建文件夹(只能创建一层)：

CreateDirectory(“D:\\a”,NULL);

CreateDirectory(“C:\\a”,”D:\\b”,NULL);

删除文件夹(只能删除空文件夹)：

RemoveDirectory(“C:\\a”);

检测逻辑驱动器：

GetLogicalDrives();

GetLogicalDriveStrings(len,buf);

获取驱动器类型(DRIVE_CDROM)：

GetDriveType(“D:\\”);

五、网络

打开网络资源枚举过程（winnetwk.h、Mpr.lib）：

WNetOpenEnum(RESOURCE_GLOBAL,RESOURCETYPE_ANY,0,NUL

L,hnet);

枚举网络资源：

WNetEnumResource(hnet,&count,pNetRsc,&size);

关闭网络资源枚举过程：

WNetCloseEnum(hnet);

打开关闭 WinSocket 库：

WSAStartup(version,&wsa);

WSACleanup();

创建套接字：

socket(AF_INET,SOCK_STREAM,IPPROTO_TCP);

绑定套接字 IP 和端口：

bind(sock,&addr,len);

监听 TCP 连接：

listen(sock,10);

接收 TCP 连接请求：

accept(sock,&addr,&len);

客户端连接：

connect(sock,&addr,len);

发送 TCP 数据：

send(sock,buf,len,0);

接收 TCP 数据：

recv(sock,buf,len,0);

发送 UDP 数据：

sendto(sock,buf,len,0,&addr,len);

接收 UDP 数据：

recvfrom(sock,buf,len,0,&addr,&len);

六、服务

打开 SCM 服务控制管理器：

OpenSCManager(NULL,NULL,SC_MANAGER_ALL_ACCESS);

创建服务：

CreateService(mgr,"MyService","
MyService",SERVICE_ALL_ACCESS,SERVICE_WIN32_OWN_PROCE
SS,SERVICE_AUTO_START,SERVICE_ERROR_IGNORE,path,NULL,N
ULL,NULL,NULL,NULL);

打开服务对象：

OpenService(mgr," MyService ",SERVICE_START);

启动服务：

StartService(serv,0,NULL);

查询服务状态：

QueryServiceStatus(serv,&state);

关闭服务句柄：

CloseServiceHandle(hdl);

连接到 SCM：

StartServiceCtrlDispatcher(DispatchTable);

注册服务控制函数：

RegisterServiceCtrlHandler("MyServicer",ServiceCtrl);

设置服务状态：

SetServiceStatus(hss,&ServiceStatus);

控制服务：

ControlService(serv,SERVICE_CONTROL_STOP,&state);

删除服务：

DeleteService(serv);

遍历服务：

EnumServicesStatus(hscm,SERVICE_WIN32|SERVICE_DRIVER,SERVICE_STATE_ALL,&srvSts,len,&size,&count,NULL);

查询服务配置：

QueryServiceConfig(hserv,&srvcfg,size,&size);

七、消息

发送消息：

SendMessage(HWND_BROADCAST,WM_LBUTTONDOWN,0,0);

接收消息：

GetMessage(&msg,NULL,0,0);

投递消息：

PostMessage(HWND_BROADCAST,WM_LBUTTONDOWN,0,0);

获取消息：

PeekMessage(&msg,NULL,0,0);

转换消息：

TranslateMessage (&msg);

分发消息：

DispatchMessage (&msg);

等待消息：

WaitMessage();

发送退出消息：

PostQuitMessage(0);

安装消息钩子：

SetWindowsHookEx(WH_KEYBOARD,keyBoardProc,0,tid);

卸载消息钩子：

```
UnhookWindowsHookEx(hhk);
```