



PAraDOx CONFerence 2009

The Way of Inject code to Process

2009.1.15

김 경 수

(kaspvxx@naver.com)

인하대학교 IGRUS



시작 하며...

이번 시간에는 DLL Injection 기법을 이용해서 다른 프로세스에 자신의 코드를 삽입하는 방법을 배웁니다.

실제로 이 기법은 간단하면서도 매우 강력하고 많은 분야에 응용되어질수 있습니다.

이용하는 API함수는 CreateRemoteThread 와 LoadLibrary 함수를 사용할 것이며 이를 이용한 게임 해킹 기법을 다룰 것입니다.

목 차

1. What is DLL Injection?
2. DLL Injection 이 왜 가능한가?
3. 방법을 알려다오.
4. 프로세스 공간에 DLL 침투 시키기
5. 실제 구현해보자.
6. 지뢰찾기 액 만들기
7. 실전 맵액 제작 해보기
8. 활용 시연 동영상
9. 참고 자료 / Q&A ?

1.What is DLL Injection?

DLL Injection 이 왜 가능한가?

방법을 알려다오.

프로세스 공간에 DLL 침투 시키기

실제 구현해보자.

지뢰찾기 핵 만들기

실전 맵핵 제작 해보기

활용 시연 동영상>

참고 자료 / Q&A ?

1. What is DLL Injection ?

DLL Injection이란 윈도우의 동적 라이브러리 (Dynamic Linking Libray)로 윈도우 환경에서 윈도우의 파일은 하나만 달랑 실행되는게 아닙니다. 내부적으로 많은 DLL을 로딩해야 하는데 이 DLL을 강제로 주입한다는 것입니다.

엄격히 말해 DLL을 주입함으로써 프로세스가 스스로 코드를 변조시키게끔 유도하는 것입니다.



각 DLL은 프로세스의 보호 메모리 주소 공간에 맵핑되며 DLL에 우리가 원하는 루틴을 주입하면 동일하게 실행 프로그램 내에도 적용된다

What is DLL Injection?

2. DLL Injection이 왜 가능한가 ?

**방법을 알려다오.
프로세스 공간에 DLL 침투 시키기
실제 구현해보자.
지뢰찾기 핵 만들기
실전 맵핵 제작 해보기
활용 시연 동영상>
참고 자료 / Q&A ?**

2. DLL Injection이 왜 가능한가?

1. 다른 프로세스에 메모리 공간을 쓸수 있다.
2. 다른 프로세스에 함수를 실행 시킬수 있다.
3. 런타임에 동적으로 DLL의 로딩이 가능하다
4. DLL이 로드될때 DllMain() 함수가 실행된다
5. DLL이 타겟 프로세스의 보호 메모리 주소에 맵핑된다!

What is DLL Injection?
DLL Injection 이 왜 가능한가?

3. 방법을 알려다오

프로세스 공간에 DLL 침투 시키기
실제 구현해보자.
지뢰찾기 핵 만들기
실전 맵핵 제작 해보기
활용 시연 동영상>
참고 자료 / Q&A ?

3. 방법을 알려다오

Windows에서는 동적으로 DLL을 로딩 시켜주는 함수가 있습니다. 바로 LoadLibrary() 함수요, 바로 이 함수를 타겟 프로세스에서 실행할 수 있게 만들어주면 됩니다.

LoadLibrary() 함수를 실행시키기 위해 이 함수를 CreateRemoteThread() 함수에게 인자를 넘겨주면 장땡임, 그러나 이를 구현하기 위한 몇 가지 과정을 살펴봅시다.

P s e u d o C o d e

1. 주입할 원격 프로세스에 대한 핸들을 얻는다 (Open Process)
2. 원격 프로세스 내에서 DLL이름을 위한 메모리를 할당한다.(VirtualAllocEx)
3. 할당된 메모리에 전체 경로를 포함한 DLL이름을 기록한다(WriteProcessMemory)
4. CreateRemoteThread와 LoadLibrary를 이용하여 원격프로세스에 DLL을 주입한다.
5. 원격 프로세스가 종료할때까지 기다린다 (WaitForSingleObject).
6. 원격 쓰레드의 탈출 코드를 얻는다 (GetExitCodeThread)

What is DLL Injection?
DLL Injection 이 왜 가능한가?
방법을 알려다오.

4 . 프로세스공간에 DLL 침투시키기

실제 구현해보자.
지뢰찾기 핵 만들기
실전 맵핵 제작 해보기
활용 시연 동영상>
참고 자료 / Q&A ?

실행 프로그램(PE Format)

IMAGE_IMPORT_DESCRIPTOR

Kernel32.dll

User32.dll

GDI32.dll

기타 여러 dll...

VirtualAllocEx()
타겟 프로세스에 메모리공간 대여

MyDll.dll
(우리가 구현한 dll)

실행 프로그램(PE Format)

IMAGE_IMPORT_DESCRIPTOR

Kernel32.dll

User32.dll

GDI32.dll

기타 여러 dll...

WriteProcessMemory()로
우리가 주입할 dll의 경로를 써줌
(우리가 실행시킬 명령)

MyDll.dll
(우리가 구현한 dll)

실행 프로그램(PE Format)

IMAGE_IMPORT_DESCRIPTOR

Kernel32.dll

User32.dll

GDI32.dll

기타 여러 dll...

CreateRemoteThread로
LoadLibrary함수를 인자로 받는
함수를 실행함

LoadLibraryA 인자에
MyDll.dll의 경로를 넘겨줌

MyDll.dll
(우리가 구현한 dll)

실행 프로그램(PE Format)

IMAGE_IMPORT_DESCRIPTOR

Kernel32.dll

User32.dll

GDI32.dll

기타 여러 dll...

MyDll.dll

프로세스 보호 메모리 주소에
맵핑된 DLL

What is DLL Injection?
DLL Injection 이 왜 가능한가?
방법을 알려다오.
프로세스 공간에 DLL 침투 시키기

5. 실제 구현해보자.

지리찾기 핵 만들기
실전 맵핵 제작 해보기
활용 시연 동영상>
참고 자료 / Q&A ?

5. 실제 구현해 보자.

1. FindWindow()를 통해 주입하고자 하는 프로세스의 핸들을 얻어옵니다.
2. GetWindowThreadProcessId()를 통해 타겟 프로세스의 스레드를 얻어옵니다.
3. OpenProcess() 함수를 이용해 타겟 프로세스의 접근 권한을 얻어옵니다.
4. VirtualAllocEx()함수를 이용해 메모리 공간을 얻어온후
5. WriteProcess()함수를 사용하여 메모리를 기록하고 그공간에 LoadLibrary()인자를 주입하는 CreateRemoteThread()함수를 실행시켜 줍니다.

What is DLL Injection?
DLL Injection 이 왜 가능한가?
방법을 알려다오.
프로세스 공간에 DLL 침투 시키기
실제 구현해보자.

6. 지뢰찾기 핵 만들기

실전 맵핵 제작 해보기
활용 시연 동영상>
참고 자료 / Q&A ?

예제 소스 1.

```
hProcess_A = FindWindow(NULL, "지뢰 찾기"); // 1
```

```
if(!hProcess_A){  
    MessageBox(HWND_DESKTOP, "윈도우를 찾을 수 없습니다.", "오류", MB_OK);  
    return 0;  
}  
else  
    MessageBox(HWND_DESKTOP, "윈도우 찾기 성공!", "알림", MB_OK);
```

```
nThreadID = GetWindowThreadProcessId((HWND)hProcess_A, &nPID); // 2
```

```
hProcess = OpenProcess(PROCESS_ALL_ACCESS, FALSE, nPID); // 3
```

// 새로운 메모리 공간 할당

```
char *pszParam = NULL;  
pszParam = (char*) VirtualAllocEx(hProcess, NULL, MAX_PATH + _MAX_FNAME,  
                                   MEM_COMMIT, PAGE_READWRITE); //4
```

```
if(pszParam == NULL)  
{  
    MessageBox(0, "메모리 할당 실패", "윌더백", MB_OK);  
    return 0;  
}
```

```
WriteProcessMemory(hProcess, pszParam, (void*)szLibPath, strlen(szLibPath), NULL); //5
```

예제 소스 2.

```
void *pAddr_Of_LoadLibrary;

hModule = LoadLibrary("kernel32.dll"); //6

pAddr_Of_LoadLibrary = (void *)GetProcAddress(hModule, "LoadLibraryA"); // 7

HMODULE hKernel32 = GetModuleHandle("Kernel32"); //8

if ( hKernel32 == NULL)
{
    MessageBox(HWND_DESKTOP,"커널 주소가 없음","HUL",MB_OK);
    return 0;
}

HMODULE ( __stdcall *pLoadLibrary)(LPCTSTR);

pLoadLibrary = (HMODULE( __stdcall*)(LPCTSTR))GetProcAddress(hKernel32,"LoadLibraryA"); //9

if ( pLoadLibrary == NULL)
{
    MessageBox(HWND_DESKTOP,"pLoadLibrary 주소!","HUL",MB_OK);
    return 0;
}

if( CreateRemoteThread(hProcess,NULL,0,(LPTHREAD_START_ROUTINE)pLoadLibrary,pszParam,0,&nThreadID) //10
== NULL) {
    MessageBox(HWND_DESKTOP,"CreateRemoteThread Error","안돼~",MB_OK);
}
```

예제 소스 3.

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

BOOL APIENTRY DllMain(HANDLE hModule, DWORD ul_reason_for_call, LPVOID lpReserved)
{
    BOOL hResult;

    char szBuffer[256];

    switch(ul_reason_for_call)
    {
        case DLL_PROCESS_ATTACH:
            sprintf(szBuffer, "PID:%d 나는 지뢰찾기지령~ DLL.dll loaded", GetCurrentProcessId());
            MessageBox(NULL, szBuffer, "확인", MB_OK);

            break;

        case DLL_PROCESS_DETACH:
            sprintf(szBuffer, "PID:%d Dll.dll unloaded!", GetCurrentProcessId());
            MessageBox(NULL, szBuffer, "확인", MB_OK);

            break;
    }
    return TRUE;
}
```

DLL Injection이 성공적으로 이뤄진 모습

The screenshot displays the Process Explorer window from Sysinternals. The 'Process' list shows several running processes, with **winmine.exe** highlighted by a red rectangle. A small dialog box titled '확인' (Confirm) is overlaid on the process list, displaying the text: 'PID:592 나는 지뢰찾기지령~ DLL,dll loaded' and a '확인' button. Below the process list, the 'DLL' list is visible, showing various system DLLs. The **Dll.dll** entry is highlighted by a red rectangle. The status bar at the bottom indicates CPU Usage: 5,22%, Commit Charge: 12,62%, Processes: 64, and Physical Usage: 23,07%.

Process	PID	CPU	Description	Company Name
mspaint.exe	1084		Paint	Microsoft Corporation
winmine.exe	756	0,75	Entertainment Pack Min...	Microsoft Corporation
procexp.exe	528	3,73	Sysinternals Process E...	Sysinternals - www,s...
psqltray.exe	1832		Fingerprint Tray Applica...	UPEK Inc,
BWAutorun.exe	336		USB Autorun	Bitwire corp,
tabbrowsingup.exe	340			
ent,aye	576		ALYac Agent for Free	ESTsoft Corp
nt30.exe	3444		Inciter v3,0 Client Module	SoftRun Inc,
fy.exe	3456		Inciter 2006 ICNotify mod...	SoftRun Inc,
e.exe	2124		Console IME	Microsoft Corporation

Name	Description	Company Name	Vers
AcGenral,DLL	Windows Compatibility DLL	Microsoft Corporation	5,01,2
ADVAPI32,dll	Advanced Windows 32 Base API	Microsoft Corporation	5,01,2
COMCTL32,dll	User Experience Controls Library	Microsoft Corporation	6,00,2
ctypes.nls			
Dll.dll			
GDI32,dll	GDI Client DLL	Microsoft Corporation	5,01,2
imekr70,ime	Microsoft IME 2003	Microsoft Corporation	7,00,5
IMM32,DLL	Windows XP IMM32 API Client DLL	Microsoft Corporation	5,01,2
kernel32,dll	Windows NT BASE API Client DLL	Microsoft Corporation	5,01,2
locale,nls			
LPK,DLL	Language Pack	Microsoft Corporation	5,01,2
MGKBHook,dll			
MSACM32,dll	Microsoft ACM Audio Filter	Microsoft Corporation	5,01,2
MSCTF.dll	MSCTF Server DLL	Microsoft Corporation	5,01,2

CPU Usage: 5,22% Commit Charge: 12,62% Processes: 64 Physical Usage: 23,07%

코드 후킹 해서 맵핵 제작해보기

Paused

LEMTWHC / KBR...S

지뢰를 출력 이전에 eax 레지스터에 0A를 저장 -> 01002F80
지뢰를 밟았을때 출력해주는 주소

01002F74	83C2 20	add edx, 20
01002F77	49	dec ecx
01002F78	75 EA	jnz short winmine,01002F64
01002F7A	5F	pop edi
01002F7B	5B	pop ebx
01002F7C	5E	pop esi
01002F7D	C2 0800	retn 8
01002F80	A1 3853000	mov eax, dword ptr ds:[1005338]
01002F85	83F8 01	cmp eax, 1
01002F88	7C 4E	j short winmine,01002FD8
01002F8A	53	push ebx
01002F8B	56	push esi
01002F8C	8B35 3453000	mov esi, dword ptr ds:[1005334]
01002F92	57	push edi
01002F93	BF 60530001	mov edi, winmine,01005360
01002F98	8B00	mov edx, eax
01002F9A	33C9	xor ecx, ecx
01002F9C	41	inc ecx
01002F9D	3BF1	cmp esi, ecx
01002F9F	7C 2E	j short winmine,01002FCF
01002FA1	8A040F	mov al, byte ptr ds:[edi+ecx]
01002FA4	A8 40	test al, 40

Registers (FPU)

EAX 0000000A

ECX 0007FD10

EDX 7C93E4F4 ntdll,KiFastSystemCallRet

EBX 00000001

ESP 0007FD34 ASCII "?"

EBP 0007FDB4

ESI 00000000

EDI 00000000

EIP 01002F80 winmine,01002F80

C 0 ES 0023 32bit 0(FFFFFFFF)

P 1 CS 001B 32bit 0(FFFFFFFF)

A 0 SS 0023 32bit 0(FFFFFFFF)

Z 1 DS 0023 32bit 0(FFFFFFFF)

S 0 FS 003B 32bit 7FFDE000(FFF)

T 0 GS 0000 NULL

D 0

O 0 LastErr ERROR_SUCCESS (00000000)

EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty -UNORM BCE0 01050104 00000000

예제 소스 4.

```
BOOL __fastcall Intercept(void);
```

```
BOOL APIENTRY DllMain(HANDLE hModule, DWORD ul_reason_for_call, LPVOID lpReserved)
{
    BOOL hResult;

    char szBuffer[256];

    switch(ul_reason_for_call)
    {
    case DLL_PROCESS_ATTACH:
        // MessageBox(NULL, szBuffer, "확인", MB_OK);
        hResult = Intercept();
        break;

    case DLL_PROCESS_DETACH:
        sprintf(szBuffer, "PID:%d Dll.dll unloaded!", GetCurrentProcessId());
        MessageBox(NULL, szBuffer, "확인", MB_OK);
        break;
    }
    return TRUE;
}
```

예제 소스 5.

```
BOOL __fastcall Intercept(void)    //이 함수는 DLL이 주입될때 실행되며 실행되며
{                                  // 인젝션 되는 시점에서 모든 흐름이 이함수로 넘어오게 됩니다.
    __asm{

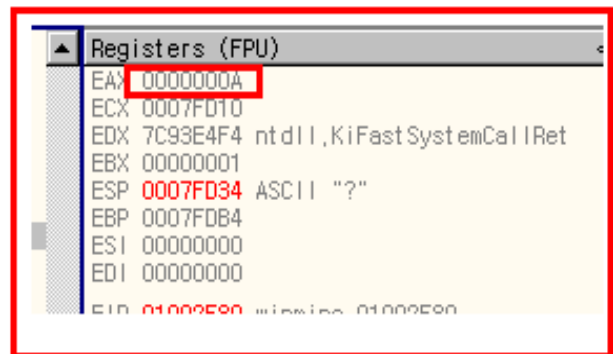
        pushad // 그렇기 때문에 현재 상태의 모든 레지스터 백업

        xor ebx,ebx //나머지 레지스터는 0으로 초기화
        xor edx,edx
        xor esi,esi
        xor edi,edi

        mov eax,0x0000000A // 지뢰를 출력 해주는 함수를 부르기전에 0x0A를 저장
        mov ecx,0x01002F80 //지뢰를 출력해주는 주소

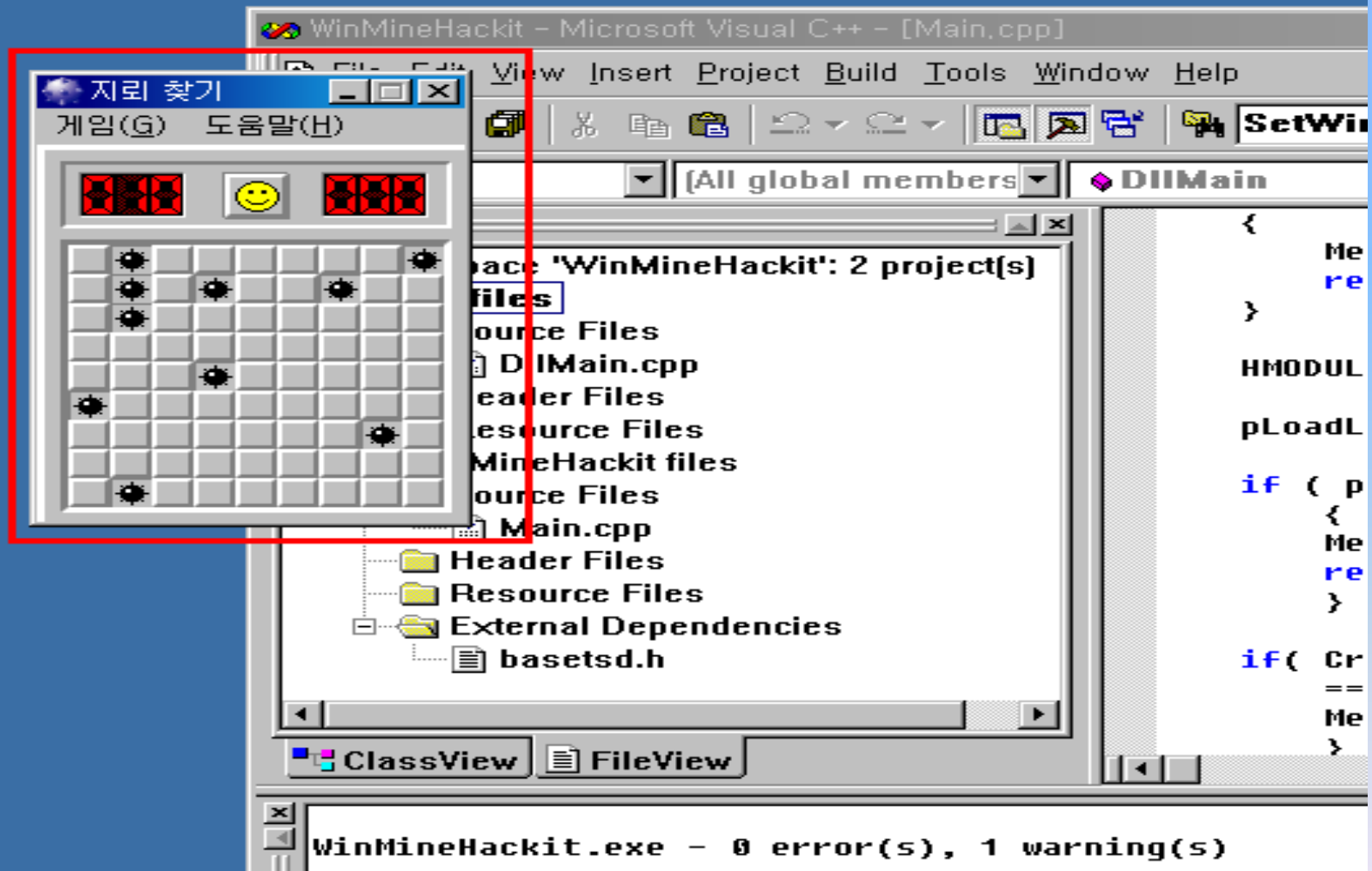
        push eax // 지뢰를 모두 출력 해주기 이전에 eax를 push
        call ecx //그리고 0x01002F80 을 호출

        nop
        nop
        popad // 스택 복구
    }
    return TRUE;
}
```



Registers (FPU)	
EAX	0000000A
ECX	0007FD10
EDX	7C93E4F4 ntdll.KiFastSystemCallRet
EBX	00000001
ESP	0007FD34 ASCII "?"
EBP	0007FD84
ESI	00000000
EDI	00000000
EIP	01002F80 winapi_01002F80

DLL Injection을 통한 맵핵을 제작한 모습



What is DLL Injection?
DLL Injection 이 왜 가능한가?
방법을 알려다오.
프로세스 공간에 DLL 침투 시키기
실제 구현해보자.
지뢰찾기 핵 만들기

6. 실전 맵핵 제작 해보기

활용 시연 동영상>
참고 자료 / Q&A ?

6. 실전 맵핵 제작 해보기.

이번에 우리는 세계적인 모 게임 제작회사 발리자드사의 게임 워크래프트삼을 후킹하여 직접 맵핵을 제작하는 내용을 대략적으로 보여드릴 것입니다.

위에서 소개한 내용처럼 CreateRemoteThread와 LoadLibrary 테크닉을 사용할 것이며, 워크래프트삼을 맵핵을 제작하기 위한 핵심 내용만을 다루겠습니다.

워크래프트삼의 중요 DLL

Mwarcraft iii

IMAGE_IMPORT_DESCRIPTOR

4기가 바이트 가상(보호) 메모리 주소

game.dll

game.dll : 주로 게임정보들을 다루는 중요 dll, 예를들어 게임의 시야 내부 정보 처리 등을 담당

storm.dll

storm.dll : 주로 그래픽결한 함수를 담당하는 (예를들어 Skill) 중요 dll

중요 함수는 dll 내부에서 export 시키지 않고 사용하여 분석을 어렵게 만듦

워크래프트삼에 대한 디버깅 특권 권한얻기

```
HANDLE hcurrent=GetCurrentProcess();
HANDLE hToken;

BOOL bret=OpenProcessToken(hcurrent,40,&hToken);

LUID luid;

bret=LookupPrivilegeValue(NULL,"SeDebugPrivilege",&luid);
TOKEN_PRIVILEGES NewState,PreviousState;
DWORD ReturnLength;
NewState.PrivilegeCount =1;
NewState.Privileges[0].Luid =luid;
NewState.Privileges[0].Attributes=2;
bret=AdjustTokenPrivileges(hToken,FALSE,&NewState,28,&PreviousState,&ReturnLength);

if ( bret == FALSE )
{
    printf("디버깅 특권 권한 오픈 실패!\n");

#ifdef _DEBUG
    DbgPrintf("디버깅 특권 권한 오픈 실패!\n");
#endif

    return FALSE;
}

#ifdef _DEBUG
DbgPrintf("<디버깅 특권 권한 오픈!>\n");
#endif
```

주요 시야 정보 1 리버싱 데이터

```
mov     edx, [esp+18h+arg_4]
movzx   esi, word ptr [edx]
movzx   ecx, si
movzx   edx, ax
and     ecx, edx      ; 2바이트를 Nop 로밀어버리자!
jnz     short loc_6F3A04E1
```

```
mov     ecx, [esp+18h+var_4]
push    eax
mov     eax, [esp+1Ch+var_8]
push    eax
push    ecx
mov     ecx, [edi+34h]
call    sub_6F00DEF0
mov     edx, eax
shr     edx, 2
and     edx, 1
jnz     short loc_6F3A04E1
```

위의 어셈코드를 패치하면 시야에
가려져있는 유닛을 볼수 있습니다.

game.dll 안에 구현되어있습니다.

주요 시야 정보 2 리버싱 데이터

```
.text      push    esi
.text      call    sub_6F3A0500
.text      and     ebx, eax      ; 같은 좌표상에 있는 유닛인가
.text      ;
.text      mov     [esp+1Ch], eax
.text      jnz     short loc_6F398E19
.text      cmp     [edi+328h], ebp ; 투명 유닛이라면
.text      jnz     short loc_6F398E25
.text
loc_6F398E19:      ; CODE XREF: sub_6F398D10+FF1j
.text      mov     edx, [esi]
.text      mov     eax, [edx+1A8h]
.text      mov     ecx, esi
.text      call    eax
.text
loc_6F398E25:      ; CODE XREF: sub_6F398D10+1071j
.text      push    1
.text      lea     ecx, [esp+38h] ; 시야를 보이지 않게 점프
.text      push    ecx
.text      mov     ecx, esi
.text      call    sub_45277400
```

위의 어셈코드를 패치하면 투명유닛 또한 볼수 있습니다.

주요 시야 정보 3 리버싱 데이터

.text:	mov	eax, 1	<- eax을 0으로 바꾸면 미니맵에서도 시야에 가려진 유닛을 볼수 있습니다.
.text:	shr	eax, cl	
.text:	cmp	eax, ebx	
.text:	mov	[esp+30h], eax	
.text:	jnz	short loc_6F3608B2	
.text:			

game.dll 의 주소값 얻어오기

```
DWORD getDllBaseAddr(TCHAR *szDllName, DWORD nPID)
{
    HANDLE hSnap32Handle;

    MODULEENTRY32 me;

    hSnap32Handle = CreateToolhelp32Snapshot(TH32CS_SNAPMODULE, nPID);

    me.dwSize = sizeof(MODULEENTRY32);

    if (Module32First(hSnap32Handle, &me)){
        while(Module32Next(hSnap32Handle, &me)) {

            if (strcmp(szDllName, me.szModule) == 0){
                CloseHandle(hSnap32Handle);
                return (DWORD) me.modBaseAddr;
            }

        }
    }

    CloseHandle(hSnap32Handle);

    return 0;
}
```

워크래프트삼의 코드 변조 금지 해제

```
void* codePatch(void* dest, void* source, int nSize){  
  
    DWORD oldsProt,olddProt;  
  
    VirtualProtect((char*)dest,nSize,PAGE_EXECUTE_READWRITE,&oldsProt);  
    VirtualProtect((char*)source,nSize,PAGE_EXECUTE_READWRITE,&olddProt);  
    memcpy((char*)dest,(char*)source,nSize);  
    VirtualProtect((char*)source,nSize,olddProt,&olddProt);  
    VirtualProtect((char*)dest,nSize,oldsProt,&oldsProt);  
  
    return (void*)dest;  
}
```

기계어 코드를 주입하여 패치 시키기

```
#ifdef _DEBUG
DbgPrintf("<Storm.dll 주소 = %08X>%n", GAMEINFORM.stormBase);
#endif
```

```
DWORD gamemapAdr= GAMEINFORM.gameBase;
```

```
codePatch((int*)(0x3A04AB+gamemapAdr), "wx90wx90", 2);

codePatch((int*)(0x36087c+gamemapAdr), "wx00", 1);

codePatch((int*)(0x28464C+gamemapAdr), "wx90wx90", 2);

codePatch((int*)(0x284662+gamemapAdr), "wxEBwx29", 2);

codePatch((int*)(0x281F1C+gamemapAdr), "wx40wx3", 2);

codePatch((int*)(0x73B949+gamemapAdr), "wxB2wx00wx90wx90wx90wx90", 6);

codePatch((int*)(0x398E01+gamemapAdr), "wx90wx90wx90wx90wx90wx33wx0wx40", 8);

codePatch((int*)(0x360C91+gamemapAdr), "wx3Bwx0wx0Fwx85wx30wx04wx00wx00", 8);

codePatch((int*)(0x3558FE+gamemapAdr), "wx90wx90wx90", 3);

return TRUE;
```

What is DLL Injection?
DLL Injection 이 왜 가능한가?
방법을 알려다오.
프로세스 공간에 DLL 침투 시키기
프로세스공간에 DLL침투시키기
실제 구현해보자.
지뢰찾기 핵 만들기
실전 맵핵 제작 해보기

7. 활용 시연 동영상

참고 자료 / Q&A ?

7. 활용 시연 동영상



What is DLL Injection?
DLL Injection 이 왜 가능한가?
방법을 알려다오.
프로세스 공간에 DLL 침투 시키기
프로세스공간에 DLL침투시키기
실제 구현해보자.
지뢰찾기 핵 만들기
실전 맵핵 제작 해보기
활용 시연 동영상

참고 자료 / Q&A ?

8. 참고자료 / Q&A

1. Art of Hooking – AmesianX
2. 해킹, 파괴의 광학 – 김성우
3. Windows Programming 고급편 – 최호성
4. DLL Injection – Proxima
5. 윈도우 환경에서의 메모리 인젝션 기술과 인젝션 된 DLL 분석 기술 – 황현욱, 채종호, 윤영태