Subject: Re: Function Hooking
Posted by Neijwiert on Thu, 25 Dec 2014 00:41:32 GMT
View Forum Message <> Reply to Message

jonwil wrote on Wed, 24 December 2014 14:42If you want to hook Commands->Find_Object, just read the address out of that variable (the "original" Find_Object) then replace it with the address of your new function.
Your new funxtion would then call the stock function through the pointer you saved ealier.
p

That would result in an infinite loop? For example:

```
typedef GameObject *(** FindObjectPointer)(int);
FindObjectPointer OriginalFindObject;
GameObject *Find_Object_Test(int obj_id)
{
 Console_Output("Finding object with id: %d\n", obj_id);

 return (* OriginalFindObject)(obj_id);
}

TimeMachine::TimeMachine()
{
 OriginalFindObject = &Commands->Find_Object;
 *&Commands->Find_Object = &Find_Object_Test;
}
```

The OriginalFindObject would point right back to the hooked one. Im trying to catch all calls to the original method and then do some stuff. I'm just using Find_Object as an example, the actual command I'm going to target is Start_Timer.

When I compile and run this I get an infinite loop.

NOTE: I'm also trying to catch calls to the method outside of my DLL. So there's no other way than memory hooking it with a jump? Or am I just thinking to difficult right now?