If you have to cast, you can't afford it

devblogs.microsoft.com/oldnewthing/20091023-00

October 23, 2009

Raymond Chen

A customer reported a crash inside a function we'll call `XyzConnect`:

```
DWORD XyzConnect(
    __in DWORD ConnectionType,
    __in PCWSTR Server,
    __in PCWSTR Target,
    __out void **Handle);
...
// HACK - Create a dummy structure to pass to the XyzConnect
// function to avoid AV within the function.
int dummy = 0;
if ( NO_ERROR != ( XyzConnect( 0, L"", L"", (PVOID*)&dummy ) )
{
    TRACE( L"XyzConnect failed." );
    return FALSE;
}
...
```

The title of today's entry gives the answer away. (The title is also an exaggeration, but it's a pun on the saying *If you have to ask, you can't afford it*.)

The last parameter to the `XyzConnect` function is declared as a `void**`: A pointer to a generic pointer. Note that it is not itself a generic pointer, however. A generic pointer can point to anything, possibly unaligned. But this is an aligned pointer to a generic pointer. Therefore, the memory for the generic pointer must be aligned in a manner appropriate to its type.

But this caller didn't pass a pointer to a pointer; the caller passed a pointer to an `int`, and an `int` has different alignment requirements from a pointer on 64-bit systems. (You might conclude that this decision was the stupidest decision on the face of the planet, but that's a different argument for a different time. For example, I can think of decisions far stupider.)

When the `XyzConnect` function tries to dereference this purported `void **` pointer, it encounters an alignment fault, because it does not in fact point to a `void *` as the type claims, but rather points to a `DWORD`. A `DWORD` requires only 32-bit alignment, so you have a 50% chance that the `DWORD*` is not suitably aligned to be a `void*`.

Mind you, you also have a 100% chance of a buffer overflow, because a `DWORD` is only four bytes, whereas a `void*` is eight bytes. The function is going to write eight bytes into your four-byte buffer.

When this question was posed, one person suggested changing the `DWORD` to a `__int64`, since the `__int64` is an 8-byte value, which is big enough to hold a pointer on both 32-bit and 64-bit Windows. Then again, it's overkill on 32-bit systems, since you allocated eight bytes when you only needed four. Another suggestion was to use `DWORD_PTR`, since that type changes in size to match the size of a `void*`.

Well, yeah, but here's another type that matches the size of a `void*`: It's called `void*`.

Just declare `void *dummy` and get rid of the cast. And get rid of the comment while you're at it. If you do it right, you don't need the cast or the hack.

```
void *handle = 0;
if ( NO_ERROR != ( XyzConnect( 0, L"", L"", &handle ) )
{
    TRACE( L"XyzConnect failed." );
    return FALSE;
}
```

A large number of porting problems can be traced to incorrect casts. The original author probably inserted the cast to "shut up the compiler" but the compiler was trying to tell you something.

Any time you see a function cast or see a cast to/from something other than `void*` or `BYTE*`, then you should be suspicious, because there's a chance somebody is simply trying to shut up the compiler.

Raymond Chen

Follow