How can I generate a stack backtrace that is independent of ASLR?

devblogs.microsoft.com/oldnewthing/20160525-00

May 25, 2016

Raymond Chen

When you capture a stack backtrace with the `CaptureStackBacktrace` function, the addresses returned are absolute addresses. If you're capturing these values for future correlation, then saving the raw addresses is not interesting because there's no guarantee that the modules in your process will be loaded at the same address every time. And indeed, with address space layout randomization (ASLR), they will almost certainly *not* be loaded at the same address each time.

So how do you save this backtrace in a way that lets you recognize it if it happens again?

For each address in the stack backtrace, convert it to a module and an offset. You can use the `GetModuleHandleEx` function to obtain the handle to the enclosing module. This is useful for two things:

1. You can call `GetModuleFileName` to get the name of the module. You probably want to save only the file name portion and remove the directory, because the directory can vary from machine to machine.
2. You can subtract the module handle from the raw pointer, resulting in an offset.

This combination of module and offset is independent of ASLR, in the sense that if ASLR loads the module at another address, the offset of the function in the backtrace will remain the same. And from the module and offset, you can reconstruct the original stack backtrace.

You can feed the module name and offset into a hash function if you want to generate a signatore for the stack trace.

Raymond Chen

Follow