In Windows XP, even when DEP is on, it's still sometimes off

devblogs.microsoft.com/oldnewthing/20071116-00

November 16, 2007

Raymond Chen

As we saw last time, there are a variety of ways you can control DEP, one of which is to turn it on for all system processes. But even if you turn on DEP, it still sometimes turns itself off temporarily. It goes back to those bad versions of ATL.

The application compatibility team found that there were so many programs written with application frameworks that were not DEP-compatible (ATL mostly, but a few others) that nobody would actually enable DEP because the odds were close to 100% that there would be some program on the system that was not DEP-ready. Even DEP-fan Leo Davidson runs a couple of programs that don't work with DEP enabled. And it takes only one program to foul an upgrade.

When the kernel encounters a DEP exception, it checks whether thunk emulation is enabled, and if so (which it usually is), it checks whether the code sequence is one of the "well-known DEP-violating thunks". If so, then it simulates the actions the thunks would have performed and resumes execution instead of raising the exception. For example, if thunk emulation is enabled and you just took a DEP exception on the code sequence

```
mov ecx, 12345678
jmp 43218765
```

the kernel thunk emulator will perform the moral equivalent of

```
pContext->Ecx = 0x12345678;
pContext->Eip = 0x43218765;
return EXCEPTION_CONTINUE_EXECUTION;
```

You can mark your program as "I am so okay with DEP that not only do I want you to enable it, but I don't even want you to do this thunk emulation stuff" by setting the `IMAGE_DLLCHARACTERISTICS_NX_COMPAT` flag in your PE header. (The Visual Studio linker uses the `/NXCOMPAT` command line switch to set this flag.)

Raymond Chen

Follow