It rather involved being on the other side of the airtight hatchway: Defeating ASLR after you've gained RCE via ROP

devblogs.microsoft.com/oldnewthing/20241024-00

October 24, 2024

A security vulnerability report went roughly like this:

> An attacker can leverage a write-what-were vulnerability to bypass ASLR. For demonstration purposes, assume that the attached DLL is present in the process. (In practice, the attacker would rely on some other DLL in the process, but we use the attached DLL for concreteness.) First, use the write-what-where vulnerability to build a ROP payload that targets gadgets in the attached DLL. Those gadgets can then use ⟦technique⟧ to locate the address of kernel32[1] and call functions in it.

The finder noted that the attached DLL is a stand-in for some other well-known DLL, but since those well-known DLLs are serviced regularly, the gadget locations change from build to build and would not be reliable in a proof of concept demonstration.

While the finder's technique for locating the address of kernel32 is quite clever, this is a case of starting on the other side of the airtight hatchway: In order to build the ROP payload, you need to know the address of the attached DLL. in other words, you have to have *already* bypassed ASLR!

This vulnerability report is basically saying, "If you have bypassed ASLR, then you can bypass more ASLR." While this is true, it's also not a particularly interesting statement. After all, you could just repeat the initial ASLR bypass to get more ASLR bypass rather than looking for a new ASLR bypass.

Bonus reading: A quick introduction to return address protection technologies, a defense which makes it harder for ROP attacks to succeed.

[1] Actually, the finder submitted at least eight versions of this report, just substituting other DLL names for "kernel32".