It rather involved being on the other side of this airtight hatchway: Disclosure of information you already had access to

devblogs.microsoft.com/oldnewthing/20200203-00

February 3, 2020

Raymond Chen

A number of security vulnerability reports came in of the form

> If I call the `XYZ` function and pass it a crafted buffer, the function parses the buffer incorrectly and reads beyond the end of the buffer. It then returns that invalidly-read data back to the caller. This is an information disclosure vulnerability in the `XYZ` function.

The important missing detail here is that the `XYZ` function runs in the same process as the caller.

This means that the invalidly-read data came from the same process. The process is gaining access to information disclosed from itself.

This is not particularly interesting by itself. The process already had access to the disclosed data by virtue of the fact that the `XYZ` function running inside the process was able to read it. The information did not cross a security boundary, so there is no vulnerability. If your goal was to access that data, you didn't need the `XYZ` function to do it for you. You could have just read it yourself.

This is just a case of garbage-in garbage-out. If you pass garbage to a function, don't be surprised if you receive garbage in return. Of course, if this garbage crosses a security boundary, then there is a problem. But if the garbage came from the same security realm, there's nothing interesting going on. You're just disclosing information that the caller already had access to.

Bonus chatter: If the garbage came from an untrusted source, then the boundary crossing was performed by the person who took untrusted data and used it without validation.

Bonus bonus chatter: There are some functions which can be used with untrusted data, but in general, it is the caller's responsibility to pass valid data. If the function takes a double-null-terminated string, say, and you pass something that isn't a double-null-terminated

string, well, that's on you if this results in walking off the end of the buffer and returning unrelated memory.

Raymond Chen

Follow