Clicking past the warning that you are about to cross the airtight hatchway: Vulnerable file type that you are warned about

devblogs.microsoft.com/oldnewthing/20221011-00

October 11, 2022

Raymond Chen

For some reason, there was a brief spike in security vulnerability reports related to a developer tool which has a specific file type for defining startup macros. When you ran the developer tool and installed a startup macro, the reports said that you could trigger a hang or a crash in the tool, which is a denial of service or potential arbitrary code execution.

Startup macros are already known to be dangerous because they are basically a miniature scripting language, and one of the commands in the miniature scripting language is capable of launching external processes with arbitrary command lines. It's so dangerous that if you try to open it, the program first makes you acknowledge the danger.

Security Warning
You should only import startup macros from trustworthy sources because startup macros can run arbitrary executables. Would you like to import and apply this startup macro file?
OK
Cancel
In other words, this file is *equivalent to code*.

If you want to attack somebody with this file, you don't have to play fuzzing games and prime the target address space with just the right kind of heap spray or sequence of allocations, so that you can deliver your crafted file that triggers an exploitable crash. There's no need to go to all that effort. All you have to do is put your exploit directly in the macro file as a command line!

It's like saying that you found a bug in the batch file parser that, with effort, could lead to arbitrary code execution. You already have arbitrary code execution because you're a batch file. Instead of playing sneaky games with the batch file parser, just put the command you want to run in the batch file.

The people filing security reports against the developer tool had to click “OK” to get past the warning dialog that said, “Clicking OK may lead to arbitrary code execution.” And then they were upset that there was the potential for arbitrary code execution.

Raymond Chen

Follow