# It rather involved being on the other side of this airtight hatchway: Replacing a service binary

devblogs.microsoft.com/oldnewthing/20220907-00

September 7, 2022

Raymond Chen

In the category of dubious security vulnerability, we have this report:

> I have found a security vulnerability in Windows that permits an unprivileged user to gain system privileges.
>
> - Look for existing services that runs as local system.
> - For each service, check its corresponding `C:\Program Files` subdirectory to see if the directory is writable.
> - When you find one, replace the binary with a hacked version that opens a reverse shell.
> - The next time the service starts, it will run the modified binary and open a reverse shell.
> - Connect to the reverse shell and control the system.

Yes, if you can find a system that runs as local system which sits in a world-writable directory, then you can replace it, and Windows will blindly execute it the next time it needs to start the service.

This is another example of creating an insecure system and then being surprised that it's insecure.

The attack hinges on finding a writable binary that runs with local system privileges. But anybody who does that created an insecure system: They created a binary that runs with system privileges and left it world-writable! Furthermore, everything in `C:\Program Files` defaults to "writable only by administrators", so somebody who leaves a world-writable file in the `C:\Program Files` directory must have gone out of their way to do so.

This security vulnerability report presupposes that such a misconfigured program exists and shows how it can be exploited. While it's interesting to know that such an attack is possible, it doesn't carry any security consequences for Windows. The security vulnerability is in the program installer, which installed a service insecurely.

And since the program is, so far, purely hypothetical, you haven't even found a vulnerability in any program. All you're saying is "If there is a vulnerability, then I can exploit a vulnerability."

If you put it that way, it's clear that this claim by itself is not an interesting security statement.

Raymond Chen

**Follow**