Your debugging code can be a security hole

devblogs.microsoft.com/oldnewthing/20051212-11

December 12, 2005

Raymond Chen

When you're developing your debugging code, don't forget that just because it's only for debugging doesn't mean that you can forget about security.

I remember one customer who asked (paraphrased)

> We have a service, and for testing purposes we want to be able to connect to this service and extract the private data that the service is managing, the data that normally nobody should be allowed to see. That way, we can compare it against what we think the data should be. This is just for testing purposes and will not be called during normal operation. How do you recommend we do this?

Remember that the bad guys don't care whether the code you wrote was for normal use or for diagnostic purposes. If it's there, they will attack it.

The customer went to a lot of effort to protect this internal data, making sure that none of the service operations disclose it directly, but then in a haze of "this would make debugging easier", they lost their heads and added a debugging backdoor that gives direct access to this data that they had worked so hard to protect.

It doesn't matter how much you protect the front door if you leave the service entrance wide open.

I have a printer driver that insists on creating a log file in the root of the drive. This log file, which is world-readable, contains, among other things, the URLs of every single web page I have printed. If I log on as an administrator and delete the log file, it just comes back the next time I print a document.

I assume the printer vendor created this log file for diagnostic purposes, but it also creates a security hole. Everybody on the system can see the URL of any web page that was printed by anybody else.

Raymond Chen

Follow