Is deprecation of APIs a security risk?

Lori MacVittie, 2008-05-06

Someone's been playing with the MySpace APIs and found a way to exploit some deprecated [according to MySpace] services through which "private" photos suddenly became public.

Jeremiah Grossman, chief technology officer at White Hat Security, a Web application security company, attributed it to "insufficient authorization," which he said are common on all types of Web sites, not just social-networking sites.

Jeremiah's explanation is evident if you walk through the details of the exploit. You must authenticate to MySpace by logging in - it's the authorization to ***view*** the private photos that was completely broken.

MySpace is claiming that the APIs used to obtain the private photos without proper authorization were deprecated, meaning they were being "phased" out.

Deprecation is to developers what quiescing is to database administrators and bleeding-off is to TCP-focused products. Deprecation can take years, as anyone familiar with the Java language specification can tell you. The problem is that developers know it's going to take years for deprecated APIs to be removed, despite the many warnings in source code and documentation claiming "the method may be removed at any moment without warning!" because in their experience, "at any moment" never seems to happen.

The question is ***why*** those methods were deprecated in the first place. Were they lacking authorization functionality? Apparently so, if a non-authorized user was able to use that API to obtain data without the proper authorization. And if they were deprecated because they were lacking authorization functionality, or simply didn't work as they should in terms of authorization and security, why were they merely deprecated instead of removed?

Deprecation in a Web-based API a la REST also increases the number of methods, scripts, or applications that have to be maintained and increases the potential security holes through which bad guys might be able to access private data - or worse.

Methods in APIs, particularly RESTful APIs like those offered by providers like Google, Amazon, MySpace, and Facebook that claim to protect private data should never simply be "deprecated" if they might be a potential security risk. While it's certainly painful to developers to have an API "break", it's better to make things break than risk exploitation.

So the next time MySpace or Facebook or [insert social networking site storing private data] decides to change something in its APIs it should take this advice:

Don't deprecate, ***delete***.

Imbibing: Coffee

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com