News: Do security holes demand full disclosure?

Every once in a while we need to step back and reassess the effects of the release of detailed security information and tools on the real world. And that's what happened recently at DEF CON 8.0, the annual hacking conference held in Las Vegas.

I had barely stepped off the plane in Vegas when someone asked if I had heard about Marcus Ranum's Black Hat Briefings keynote, delivered on the eve of DEF CON, about full disclosure. In the context of computer security, full disclosure means publicly releasing all details of a security vulnerability, sometimes even including exploit code.

Ranum, CEO of Network Flight Recorder, must have struck a nerve in the crowd because everyone was talking about his speech as setting the tone for the conference. A tone that some people I talked to thought was one of anger.

Talking frankly about the harms of full disclosure to such a technical security audience was gutsy and I am glad that Ranum did it. It raised important questions. How are vendors responding? Are users capable of protecting themselves? How do attackers use the full disclosure information?

There is no doubt in my mind that there are security problems that would not have been fixed for many months after discovery unless they were going to be made public, with details, by the discoverer. I have spoken with security researchers who have no intention of going public with the problems they find. When they contact software vendors with their newly found vulnerabilities they get a very different response than that which I have experienced while reporting vulnerabilities as a security researcher

for the L0pht. I have been impressed how quickly the issues I have reported to Microsoft and Lotus have been fixed. Lately they have had fixes ready in only two weeks.

So most of the large vendors seem to be on their toes. They have adjusted to the reality of free unfettered speech about security issues on the Internet. With this type of response it makes sense to report the issues first to the vendor before going public. And this is what the majority of security researchers -- amateur or professional -- do.

But even with great vendor response to full disclosure we seem to have an Internet riddled with security problems. The vulnerability life cycle is: product shipped with a latent vulnerability, vulnerability found, vendor notified, patch released, public notified, and, finally, the user patches or upgrades the software thus eliminating security problem.

There are two parts of this life cycle that full disclosure only partially effects. Full disclosure can help educate current and future software and hardware developers about potential problems by clearly documenting exactly what design or implementation flaws are found in products. But it cannot make developers learn from the mistakes of the past. It cannot force better testing. This is why, even with the free flow of information about security vulnerabilities, the same problems crop up over and over: buffer overflows, unsanitized user inputs, and poor implementation of encryption. There needs to be some incentives to not repeat the mistakes of the past.

The other part of the life cycle that full disclosure only partially effects is getting users to patch their systems. Full disclosure gets a security problem more exposure because the discoverer has an incentive to notify the press whereas the vendor usually wants to downplay the issue. But most users don't translate this directly to the need to patch their systems and many never get the proper information they need to know what to do. There needs to be better ways for users to find out if they are vulnerable

and to take action in a simple and easy way.

Those critical parts of the vulnerability life cycle, the latent problems shipped with products and users not maintaining patched systems, are the heart of the Internet security problem. These are problems that need to be attacked and solved.

Getting rid of full disclosure would only make these problems worse. Sure, as Ranum argues, there would be less script kiddies spewing Web graffiti and shutting down sites with denial of service. But that would be replaced with something far worse: attackers who can uncover their own vulnerabilities, or have the connections to pay for them. With an environment of silence these attackers could cruise through networks with impunity knowing that their vulnerability knowledge will be useful for many months.

So, instead of trying to squelch the free exchange of security research information there needs to be a concerted effort to motivate vendors to build more secure products. Products that don't ship with latent vulnerabilities that are repeats of the problems of the past. There also needs to be a concerted effort to solve the issue of getting users to quickly and easily fix their vulnerable systems. If attackers can find and exploit the security problems surely there must be a way for those on the side of good to find and fix the problems first.

Weld Pond is a research scientist working with the security firm @Stake Inc..