'Responsible Disclosure' Draft Could Have Legal Muscle

A proposal recently submitted for comment to the Internet Engineering Task Force by Steve Christey of MITRE and Chris Wysopal of @Stake would create an official standard for reporting security vulnerabilities to vendors, and for vendors to respond to any such reports. It's worth reading, because if it becomes an official Internet standard, called an "RFC", it could expose those who fail to adhere to it to legal liability for negligence or defamation.

By and large the report articulates what many in the security industry have considered to be a reasonable method of reporting security vulnerabilities. Essentially the draft recommends a process whereby a person discovering a security vulnerability would notify the vendor of the vulnerability, but not publicly disclose it.

The vendor would, after having seven days to acknowledge receipt of the disclosure, be obligated to validate the existence of the vulnerability, and resolve the vulnerability either by issuing a patch or resolution, or indicating that the vulnerability is not susceptible to any fix, and recommending remedial efforts, if any. The proposal creates a role for a "coordinator" (e.g., an ISAC or CERT) to mediate between the reporter and the vendor, and to help publicize the existence of the patch or remediation.

While such policies and procedures have been frequently followed among responsible computer security professionals, those discovering vulnerabilities have complained that vendors may have no individuals specifically designated for notification of security vulnerabilities, and may be unresponsive to reporters complaints. Meanwhile, vendors complain

that hackers frequently report or even exploit vulnerabilities before the vendor can take reasonable steps to repair them.

Liabilities Associated With Reporting

The proposal is intended to address the release-and-patch security cycle of modern software. Software vendors frequently put out code that is at best insecure, and which exposes users to serious vulnerabilities. Not only might the code itself be insecure, but particular configurations of the software, or the interface between different applications, may render the overall enterprise vulnerable to attack.

Hackers -- used in the non-pejorative sense -- complain that vendors fail to take responsibility for such vulnerabilities, particularly where they result from the interaction between one vendor's product and another, blaming the other vendor for the problem, or blaming the user for improper configuration. Code may be release to be public as a "final" version, with an implicit recognition that the consumer will act as the ultimate "beta" tester. Rather than implementing a full quality assurance program, the vendor may rely on consumers to report vulnerabilities.

Imagine buying a new car where neither the brakes, seat belts or air bags have been fully tested, and having the manufacturer tell you to report any problems with the safety systems, and they will be fixed in the subsequent model year. To make matters worse, the vendor will rely upon disclaimers of warranty in the click wrap or shrink wrap license to absolve themselves of any liability for such security vulnerabilities. Additionally, vendors do not have a centralized reporting structure -- a specific individual or set of individuals whose sole responsibility is to fix security related vulnerabilities, not only of the vendor's product itself, but of vulnerabilities associated with the use of the vendor's product.

Finally, vendors are frequently seen as unresponsive to external reports of

vulnerabilities. The reporter may not know that the vendor has received the report, and is frequently kept in the dark about the status of the investigation and repair of the vulnerability. Faced with apparent inaction on the part of the vendor, the reporter may resort to "self help" and simply make public disclosure of the vulnerability.

To Report or Not To Report

From the vendor's standpoint things are equally bleak. Vulnerability reports may be nothing more than hoaxes or rumors, the investigation of which leads to a waste of time or resources.

Fixing security vulnerabilities may not be the vendor's highest priority, particularly where the product involved is freeware or shareware, previously released and unsupported software, or otherwise unprofitable to support. The vendor may not feel it is a worthy use of limited engineering resources to fix a vulnerability that is theoretical, or represents only a minor threat.

Where the vulnerability results from a configuration issue, or interoperability between programs, the vendor may simply recommend a reconfiguration of the software, even if that results in a significant loss of performance. Likewise, the interoperation problem is likely to be blamed on the other guy's software. Finally, no vendor wants to publicly admit that a product was released with a vulnerability.

All of this brings us to the reporting dilemma.

The law generally requires people to act "reasonably." That is, a vendor owes a duty of due care to the community of people who use or rely upon the product, and makes certain warranties and representations about the product itself. For products, this generally includes a warranty that the product is "fit" for its intended use, although unless prevented by state or

federal law, some of these warranties can be modified by contract -- typically the software license itself. Even if warranties (express or implied) are waived, the vendor must still act reasonably when notified of a defect in its product.

Similarly, a person discovering a vulnerability is required to act "reasonably." Publicly reporting a fictitious vulnerability would likely subject the reporter to liability for tortuous interference with business relationships, business defamation, or other potential liability. For example, willfully disseminating false information about a vulnerability could even result in liability for securities fraud or stock manipulation if the disclosure was intended to affect the price of the company that manufactured the product. If a company publicly discloses the existence of a vulnerability in a competitor's product, this could be either the act of a good Samaritan or an unfair trade practice.

Negligence Per Se

The standard of reasonable care is a difficult one to deduce -- particularly where there is little consensus on the proper thing to do. This is especially the case for reporting of security vulnerabilities. The law distinguishes between "ordinary" negligence -- simply not adhering to a standard of care -- and "negligence per se" -- acts which are so violative of a known standard of care that they are presumed to be negligent. Typically (but not invariably) failure to comply with a law or regulation is deemed to be negligence per se -- it is negligent to drive 75 MPH in a 55 MPH zone. The law or regulation establishes the standard of care.

In this case, the "Responsible Disclosure" draft sets out recommended practices and procedures for reporting of security vulnerabilities and for vendor's responses thereto. While there are flaws in the proposal, to the extent it results from a consensus, it can be seen as establishing a standard of care. Failure therefore to adhere to this standard (just like

failure to adhere to a standard such as BSD 7799) could be deemed by a court to be negligence per se.

The development of a consensus on reporting of vulnerabilities is only one step in solving the problem. Vendors need to accept social and legal responsibility for the quality and security of their products, and for their interoperability. Vendors must make those responsible for security more visible, and more public. Indeed, searching most vendor websites, one is hard pressed to find the name of a specific individual responsible for reporting security vulnerabilities.

Because of the diversity of the reporting community, it must inevitably be recognized that groups of hackers will exploit rather than report vulnerabilities, or will publicly disclose them in order to take "credit." There is no provision in the draft that would permit or require the vendor to responsibly notify the user community (even in a confidential fashion) that a vulnerability exists before a fix or patch is available. The draft essentially absolves the vendor of liability for damages that result from exploitation of a vulnerability known to the vendor, but not yet patched. In many cases, the reasonable thing for the vendor to do would be to advise the user community to disable or discontinue use of a product or functionality pending the investigation of the exploit, but the draft has no such requirement. The implication is that failing to do this on the part of the vendor is not unreasonable.

Common sense has a tendency to trump all rules and regulations. While the decision whether or not to disclose or report a vulnerability is a difficult one, common sense should prevail. All the law requires is that we act reasonably. If only we could agree on what that was.