Addressing Shared Risk in Product Application Vulnerability Assessments

Service organizations with a bespoke application in a regulated industry have special challenges in addressing application vulnerabilities. At one vendor that hosted an application containing sensitive data, fixes were not deployed to the clients' systems in a timely fashion despite there being little technical impediment. When the service provider's risk team ultimately found the key to getting security fixes accepted, it was in a nuanced appreciation of risk—specifically, the risk of appearing negligent.

The Problem

Application vulnerabilities have both proximate and secondary risk factors. The proximate risk factors are obvious—data breaches impact the affected individuals whose personal information is compromised. But the secondary risk lies in the legal exposure to the client organization, i.e., risk for which the technology service organization—whose product allowed the breach—would be responsible. Data breaches frequently give rise to legal action, i.e., action that is often rooted in negligence. As of 2016, 75 percent of cases arising from a data breach include negligence.[1] In a legal sense, negligence is defined as "a breach of duty to take proper care."[2] Negligence can be determined with some simple questions:

- Does a duty of care exist between the parties?
- Has that duty of care been breached by the offending party?
- Has damage resulted from that breach?

The definition of "duty of care" changes based on the jurisdiction. In most of the Commonwealth, it is a three-part test. "Harm must be reasonably foreseeable as a result of the defendant's conduct, the parties must be in a relationship of proximity, and it must be fair, just and reasonable to impose liability."[3] In some US jurisdictions, the first test alone determines duty of care; in others, it is absent.[4]

While an exhaustive review of how this subject is applied in different jurisdictions is beyond the scope of this article (and, quite frankly, the author), the salient point is this: However the duty is legally defined, a service provider has a responsibility to secure information, and a breach of that responsibility opens the provider to a liability rooted in negligence. Regulators have, for years, been active in enforcing due care in the case of data breaches. The US Federal Trade Commission, for instance, speaks of filing some 60 actions against "companies that put consumers' personal data at unreasonable risk."[5] It is, therefore, imperative that the providers of an application containing any form of sensitive data in a regulated environment understand the local legal and regulatory implications.

In the case of the technology service provider in this article, relevant regulators include the Canadian

Do you have something to say about this article?

Visit the *Journal* pages of the ISACA® website *(www.isaca.org/journal)*, find the article and click on the Comments link to share your thoughts.

http://bit.ly/2tRkr5R

Michael Werneburg, CIA, PMP

Is a technology risk practitioner in Toronto, Ontario, Canada. In a 23-year career spanning three continents, he has worked with firms ranging from small start-ups to some of the world's largest financial institutions. His passion is leveraging risk to effect change across technology organizations.

Office of the Superintendent of Financial Institutions (OSFI), which considers such relationships materially important to the stability of federally regulated financial institutions.[6] Industry-specific legislation such as the US Gramm-Leach-Bliley Act (finance) and the US Health Insurance Portability and Accountability Act (HIPAA) explicitly dictate the controls and practices by which data are meant to be secured. In 2016, the attorney general's office of California clarified a specific set of controls as its standard for reasonable security.[7]

> **“Even within a service organization, it is not always easy to obtain permission to approach a client concerning necessary security fixes.”**

Leading back to an application vulnerability, regulatory requirements and service audit regimes (such as the ubiquitous Service Organization Control [SOC] 2) dictate that an application vulnerability scan is performed no less than annually. They also require that the report be shared with the client. By the time a breach has occurred, the technology provider and its clients all have these annual reports in hand. By that time, it is hard to avoid the appearance of negligence if the vulnerabilities documented in those reports have not been addressed in a timely fashion, especially when those vulnerabilities are shown to put sensitive information assets at risk.

Who Does Not Want Application Fixes?

If it sounds odd that application stakeholders would not want security fixes, it is worthwhile to look at how regulated industries behave. In this example, the service provider was active in the wealth management sector. That sector typically has a conservative approach to change, strong regulatory oversight, and—when it comes to releasing software—a focus on business features over nonfunctional factors. In such an environment, the service provider cannot dictate the nature or timing of a security release, regardless of who hosts the application.

First, it is not always easy for a service provider to explain the necessity of security fixes to the client stakeholder responsible. Client stakeholders who typically manage the relationship with a service provider and who decide on and schedule expensive test-and-release procedures may not appreciate or comprehend security fixes in the first place. Frequently, the parties making these decisions have priorities relating to functional requirements—what the application does for the organization—and are not rewarded for venturing into activities that deal with nonfunctional requirements. That makes those decision makers hard to motivate through describing security fixes in terms of abstract scenarios and recent vulnerabilities. A service organization's risk team might find themselves going to great lengths, discussing the finer points of medium-priority findings vs. high or critical. Or if they finally convince the stakeholders of the urgency of a fix, they might discover that a freeze has been introduced or that the client's budget for testing and deployment is not there.

Even within a service organization, it is not always easy to obtain permission to approach a client concerning necessary security fixes. Plenty of stakeholders within the service organization have conflicting objectives, perhaps involving delivering new features, containing support costs or managing client relationships that are in a sensitive phase. In almost all cases, the service organization views its clients' budgets as finite, and many priorities compete for the same budget and not all demands can be met. The product owner, the account representative, the overworked software development and software quality assessment teams, budget oversight, and even the support team that was burned for a failed security patch deployment years prior can stop security patches from heading to clients—and they often do.

Supposing the risk team overcomes internal resistance and finds the right parties to work with on the client side, they will still have to deal with the slow-moving nature of regulated clients. Universally, clients will only accept a release once they have conducted their own acceptance testing. Any application release testing can take a great deal of effort, scheduling and expense on their part. But security fixes, with their nonfunctional nature, can be notoriously difficult for a software quality assurance function to properly regression test, and fixes sometimes require a test environment that meticulously matches the production environment. The release process at cautious, regulated firms is, likewise, highly risk-averse and demands exhaustive release notes. And again, security fixes can be hard to explain to the satisfaction of such stakeholders—especially when it comes to proving that no unintended side effects lie dormant.

> **“But security fixes, with their nonfunctional nature, can be notoriously difficult for a software quality assurance function to properly regression test.”**

And yet, that shared risk of being found negligent after a breach does not go away on its own.

Leveraging Risk

After trying for years to use logic to schedule security fixes, the risk team finally found a way to address security using the industry's risk-averse culture in its favor. Working with the service organization's executive team, the risk team developed a three-step process that focused not on vulnerabilities and impacts, but on the underlying risk inherent in the relationship: the potential legal and regulatory impacts and the relevance of negligence to the conversation.

Implementing this three-step process began as soon as the annual third-party application vulnerability assessment report was in the service provider's hands. The three steps are:

1. Work with the service organization's application developers, the project management office and the delivery team to develop estimates of:
    - The complexity of the technical fixes
    - Possible impacts to the users from the fixes, if deployed
    - Possible schedules for fix delivery

    At the enterprise mentioned previously, this step helped ensure the buy-in of internal stakeholders. It also helped the risk team filter out issues that could not be fixed for technical reasons, false positives and issues for which fixes were already in the pipeline. And it helped the clients understand the context of the third-party report.
2. Write an interpretation of the assessment report that is rich in application context and, therefore, easy for clients to understand; include impact assessments and potential schedules; and frame the vulnerabilities in terms of the joint losses that could arise from negligence if the fixes are not addressed in a timely fashion. A custom report should go to each client featuring only those portions of the scan report that impact their version of the bespoke application. This enables client-side stakeholders other than information risk personnel to understand the issues, properly weigh priorities and encourage their active participation in the conversation as informed parties.
3. Discuss the service provider's report with each client and request a signature acknowledging the report.

It was this final step that drove home the risk to the application owner on the client side: They were being asked to acknowledge risk on behalf of their employer. Acknowledging the risk is not the same as accepting it, as the conversation that followed proved. In that conversation, the client interpreted signing the report as an act of actively seeking advice on which risk they felt they had to live with and

which should be mitigated with fixes. This led to a discussion of which fixes to prioritize and how soon the service organization could get those scheduled.

In this scenario, the risk team had normalized the process of securing security fix releases. As a result, what would follow would be a business-as-usual addition of new releases to the service organization's workload.

This process is not one that should be developed after the service has entered production. It should be enshrined in the contract between the technology service provider and its clients. Sources such as the Open Web Application Security Project (OWASP)[8] have published thorough guidance on contractual language relating to the inclusion of product security in the software development and delivery life cycle. Some of these requirements include:

- A recognition by the client that they are bound to participate in the process of approving fixes arising from application vulnerability scans
- A recognition that those fixes will be released according to an agreed-upon schedule

Doing so from the outset eliminates the objections, any ambiguity in terminology and all of the other drag experienced by the service organization.

Conclusion

The vendor in the wealth management sector discussed in this article took years to find a way to assure the release of application vulnerability scans. At issue was the culture of the sector in which it was engaged. The culture had:

- A conservative approach to change
- Strong regulatory oversight that places a heavy emphasis on third-party risk arising from technology service provider relationships
- A strong focus on business features over nonfunctional factors such as security

The risk team ultimately found a way to leverage the first characteristic against the latter two. Even in the most change-adverse environments, responsible parties realize that it is hard to justify accepting an increment of risk of being found negligent for the purpose of sparing the organization some inconvenience and routine expense associated with resolving application security issues.

Endnotes

[1] Bryan Cave LLP, *2016 Data Breach Litigation Report*, 6 April 2016, *https://www.bryancave.com/en/thought-leadership/2016-data-breach-litigation-report.html*

[2] Duhaime's Law Dictionary, "Negligence Definition," *www.duhaime.org/LegalDictionary/N/Negligence.aspx*

[3] e-lawresources.co.uk, "Negligence—Duty of Care," *http://e-lawresources.co.uk/Duty-of-care.php*

[4] *Ibid*.

[5] Federal Trade Commission, *Privacy and Data Security Update (2016),* USA, January 2017, *https://www.ftc.gov/reports/privacy-data-security-update-2016#how*

[6] Canadian Office of the Superintendent of Financial Institutions, *Outsourcing of Business Activities, Functions and Processes*, May 2001, *www.osfi-bsif.gc.ca/Eng/fi-if/rg-ro/gdn-ort/gl-ld/Pages/b10.aspx*

[7] Harris, K.; *California Data Breach Report 2012-2015*, February 2016, *https://oag.ca.gov/breachreport2016*

[8] Open Web Application Security Project, "OWASP Secure Software Contract Annex," *https://www.owasp.org/index.php/OWASP_Secure_Software_Contract_Annex*