New York Law Journal
SERVING THE BENCH AND BAR SINCE 1888
WWW.NYLJ.COM
An ALM Publication
VOLUME 255—NO. 104
WEDNESDAY, JUNE 1, 2016

OUTSIDE COUNSEL

Expert Analysis

How Not to Be Victim of a Cyber-Attack: Security Rules for Lawyers

A few months ago, my software team received the following request from an attorney: "Please create a form on my WordPress website so potential clients can upload scans of important documents."

He thought it would be an improvement over the current process: Gmail, without realizing the danger of his request.

Replacing Gmail for the exchange of sensitive data was a commendable idea. Here's why: As the saying goes, if the food is free, you're on the menu. Gmail and other services (Facebook, Dropbox, Evernote) are free; their "product" is you and your data. When you first logged on and signed an initial terms of service, you turned over various rights to your data.

The terms of service for Google Drive, for example, says the company has the right to create "derivative works" from content stored there and to "publicly display and distribute such content." Gmail's terms of service says the company can analyze the content of your messages to allow advertisers to target you based on this information. What might they use the data for later? No one knows. Companies like Google can change their terms of service when they like.

By **Anna Murray**

Since software is such a popular target, we must do a better job at developing less vulnerable software.

The vignette above illustrates the critical role software development plays in law office cybersecuity. Moving to a basic WordPress form to transfer sensitive data would have been more convenient than Gmail, but it was hardly more secure, and arguably less so. Similarly insecure practices proliferate because software development is everywhere. If you put up your own blog, you're engaging in a simple form of software development. Software development encompasses all application-creation activities from whipping up WordPress blogs, to launching ecommerce websites, through configuring and installing large-scale financial systems.

Software development is efficiency-driven, deadline-driven, needs-driven and feature-driven. Being first to market with a product, especially a software product, provides such a head start over the competition they usually cannot catch up, or cannot do it easily. Achieving a first-mover advantage is critical, and in certain industries it may trump all other concerns. Twitter invented a category, called "microblogging," and quickly dominated it. Another Twitter may come along, but imagine how much effort would now be needed to eclipse the first mover.

Obviously the drive for speed and features are in direct conflict with cybersecurity considerations.

According to the U.S. Department of Homeland Security, 90 percent of cyber-attacks originate on the application layer. This means it's more common for hackers to find holes in a company's website than it is for the same malicious actors to steal credit

ANNA MURRAY *is a technology consultant and the CEO of emedia. She is the author of 'The Complete Software Project Manager: Mastering Technology from Planning to Launch and Beyond' in the Wiley CIO series. Her email address is AMurray@tmg-emedia.com.*

card numbers through a fake card reader or by installing a virus on a server. Since software is such a popular target, we must do a better job at developing less vulnerable software.

Here is one key rule to follow: Include cybersecurity thinking from the planning stage of software development all the way through launch. Your cybersecurity team will be an invaluable ally here. If you don't have one, consider outsourcing.

To follow the rule, use these essential guidelines:

1. Questionnaire for Developers. Send a cybersecurity questionnaire to the software development agency. It can be a simple process. For example, OWASP (the Open Web Application Security Project) is a nonprofit organization that publishes a list of the 10 most dangerous web application flaws and methods of dealing with those flaws.

Provide your developers with a list of the OWASP Top 10 and request they comment on how they are addressing each one, or explain why a certain item is not relevant to your project. Then, include an additional essay question. Ask potential firms to comment briefly on their cybersecurity experience and practices.

Depending on your business and risk considerations, you may need to expand this document to include hiring and firing practices, background checks, and controlling access to sensitive data. You may want to confirm your vendor has appropriate roles-segregation practices. Which means the systems administrator doesn't have the database password, and the database administrator doesn't have the system's password.

Simply verifying up front that your development team follows best practices helps ensure security through the life cycle of your product's development.

2. Choice of Platform. At the start of a project, you must choose what product or platform you are going to use to build your software. For example, you might license Microsoft's SharePoint to develop your intranet, or buy Adobe's Experience Manager to build your website. These products are designed for large enterprises and tend (tend is the operative word) to be more secure. Security features are built into the framework itself, and therefore many vulnerabilities are handled for you. Still, developers can alter even enterprise-grade products in ways that make them less secure. Don't assume choosing an enterprise-level product is a security panacea.

You must be especially vigilant if you are using open-source tools. Make sure your development team has all the necessary experience and qualifications for secure software development. Questions to ask: What plugins are they using? Have they been thoroughly vetted by the community and reviewed as secure?

Many companies do not choose enterprise development environments from big software firms like Microsoft, IBM or SAP. Instead, they opt for open-source. "Open-source" is a term used to describe software development platforms that are free and where programmers can access vast libraries of tools without paying a licensing fee. Some common open-source names are WordPress, Drupal, and Ruby on Rails.

In the open-source world, the developers contribute their work product back to the platform. Because of this, hundreds of thousands of community-built modules exist for the well-known open-source frameworks.

The benefits of open-source are obvious. In addition to affordability, they offer immense flexibility and range of resources. Open-source frameworks have been a godsend in terms of truly achieving feature-rich software and rapid application development. However, you must be especially vigilant if you are using open-source tools. Make sure your development team has all the necessary experience and qualifications for secure software development. Questions to ask: What plugins are they using? Have they been thoroughly vetted by the community and reviewed as secure?

Businesses must remember to update open-source platforms like WordPress for the latest patches. This step is simple and often forgotten.

3. Forms, Passwords and Entry Points. Any place where a user enters data into an application is a potential vulnerability. Hackers can use these input opportunities to tunnel in to deeper layers of the software. Or, they can access the accounts of your users through tricky combinations of social engineering and brute-force password cracking apps that are freely available.

Take a look at your forms, logins, and password-recovery practices. Best-practice guidelines are readily available. For things you don't catch, a security test will help. (See #8.)

4. Data Collection and Storage. Understand the sensitivity of the data you plan to collect. Is it personally identifiable information (PII) such as emails, phone numbers, and dates of

birth? If so, you must plan to encrypt the data in transit (as it's being transmitted from the web form to the database), and at rest (when it is simply stored in the database waiting for retrieval).

5. Integration Points. Integration powers most modern software development. An integration point is where your application exchanges data with another system, such as when you retrieve a user's account information from your customer relationship management (CRM) software and use it to show a profile page on your website.

To execute an integration, you will commonly create something called a webservice between your two systems with each system having an "endpoint."

But how do you know if the authorized system (CRM) is the one accessing your endpoint?

To ensure security, verification protocols must be put into place. One is called "IP Whitelisting." IP stands for Internet Protocol, which is essentially the web addresses of your servers. When IP Whitelisting is implemented, servers will only allow traffic from approved IP addresses.

Another method to secure integration points is a VPN to VPN connection. VPN stands for "virtual private network." More complex and sophisticated than IP whitelisting, a VPN to VPN connection creates a trusted and mutually authenticated "zone." Only servers admitted to that "zone" may talk to each other.

6. HTTPS. An ISP (Internet service provider) can look at web traffic passing through its network and tell what sites a user has visited. ISPs can (and do) sell that information to advertisers. If the ISP is hacked, that information is available to the hacker.

An alternative to the standard HTTP domain, is an HTTPS domain. Operating over a "secure socket layer," traffic to and from the domain is encrypted.

An HTTPS domain gives your users an added layer of protection. The pages visited on the site are encrypted against snooping by anyone trying to intercept the traffic, and so is any data exchange between your application and your users.

7. Temporary Environments and Backdoors. In most software development, programmers set up non-production environments. These temporary development spaces are like "sandboxes" where programmers can try things out before they "go live." These spaces are not behind firewalls, may not have encryption "turned on," or have whitelisting active. Further, your software vendor may have been given temporary access to a company's database for the purposes of setting up an integration. It is a good idea to treat these environments with care and apply security best practices there as well. Hackers don't really care if it is the production environment or the test environment that they got access to. Once in, they are likely capable of navigating to any environment they want.

Finally, once your software is launched, all these temporary environments must be retired and private access revoked. In short, from a security standpoint, decommissioning a dev environment (hardware, software, credentials, data) is as critical as launching your production application.

8. Testing. General third-party QA testing is one of the most overlooked areas of software development. In "third-party QA," you hire an independent quality-assurance team to test your software for all kinds of bugs, including security flaws. All too many companies say things like, "Our internal team will test the software."

The truth is even very technical staff members are not capable of thoroughly testing software for general bugs. "Penetration testing," which involves mimicking malicious hackers to find holes, is beyond the capability of most people. A professional QA team is the best strategy.

You probably plan to introduce new features on your application. This means testing must be incorporated as a maintenance task. There are automated tools such as BurpSuite to test basic code vulnerabilities. You may also want to include testing "by hand" at least quarterly to ensure ongoing security. Remember, hackers are constantly probing for vulnerabilities and developing exploits. Don't let your application become an easy target! And don't forget to evaluate your overall business processes, such as the use of tools like Gmail, to ensure data privacy and security.