MEND

The CISO's Guide to AppSec Innovation

Chris Lindsey

Threat actors operate by an ironclad rule: If it's important to businesses, it's important to them. And they certainly understand the crucial business role of applications: They are now the number one attack vector, while software supply chain attacks increased 650 percent in a year. Clearly, if you don't already have a modern application security program that can support today's digital world, you need to build one.

While finding and choosing the right technology is obviously important, it's only one part of the process. After three years as the head of application security at a multibillion dollar technology company, I know that application security isn't a department. It's a program that must work across the entirety of a development program, whether there are 30 programmers or 3,000. The foundation of that program is collaboration and education, listening and teaching – in two directions.

If we want developers to intrinsically use application security as part of the application development cycle, we need to understand how developers work – and they need to see why AppSec is important. Ultimately, vulnerable applications pose a business risk, and teaching that is a multi-fronted effort that reaches from new hires all the way up to the executive suite. Here are some of the things that worked for our program.

Find out what you don't know

One of the first things I wanted to build was an inventory of what applications exist, what we were building, and the AppSec-related details for each one. To do so, we created an application questionnaire, but instead of sending it out as an electronic survey, we conducted deep-dive interviews with software architects and associated team members to build a clear picture of that application from an application security perspective. We asked questions such as:

What languages does the app use?

- How does it communicate? (email, FTP. APIs, etc.)
- Does it communicate internally/externally?
- Where are authentication details stored? (database? active directory? Identity Providers (IdP) like Okta)?
- If a database is used, is the data encrypted at the field level?
- How are you securing passwords?
- Do you store credit card information--if so, are you PCI compliant?

Each meeting generally surfaced three or four items that needed to be looked at, and we then worked together to fix things. It was a whole different type of thought process about application security that was new to many, and it really helped developers understand what they actually owned as far as applications go.

> **"Hey, look, I can run Fiddler. I can do this. I can see the data.
> I can steal your keys. This is why you have to do certain things."**

Educate from the top down

Having senior management that understands the business risk of insecure software generally results in senior management that's invested in reducing that risk. As a software producer as well as consumer, my company had a double mandate. We needed to secure the software we developed and sold, and we also needed to check all the applications we used internally, from web apps to third-party software. With that in mind, we made sure to extend education to the C suite in several ways:

- **Synchronization.** Our team held a Security Sync Week every summer. The first day was geared toward company directors and managers, from the CEO down. We set expectations and gave guidelines about program goals as well as how to work with the security team.
- **Monthly reports.** I sent a monthly report to senior executives that aggregated the results of our security tools, such as SAST and SCA, into a high-level dashboard with indicators including the number of high vulnerabilities for every product in production, medium vulnerabilities in production, and the number of lines of code in production. Our CEO loved that because he didn't have that visibility before. Providing this information all the way up was basically a breath of fresh air to them.
- **Newsletters.** We sent out a newsletter every other month that explored topics of interest. For example, last year's topics focused on API security, which realistically is tied to everything.

Bake application security into developer onboarding

We employed more than 500 developers and had significant M&A activity. To make sure the entire developer population standardized on the same tools and understood the importance of application security, we integrated AppSec into their onboarding process.

It started with tools. Our application security tools were part of the tech package developers were given access to. Application security learning modules were also part of the onboarding checklist for developers.

I put together videos of us using SCA and SAST tools, together with a question and answer session after.

We followed up with a Skype or Teams call. That became part of the onboarding process, and more formal training followed as part of on-the-job education.

Show, don't tell

I spent many years as a developer, so I can vouch for this common mentality: If I found a tool that works for me, why would I need to go back and touch it unless a new feature appeared? If it works, developers think it ain't broke–even if it represents a security risk.

To change their minds, I spent a lot of time making sure they understood why we were asking certain things and that we were all in this together. I found it helpful to show them rather than hammering people with a PowerPoint presentation.

For example, at one Security Sync week, we did a video about brute-forcing passwords, and we actually showed them how to do it by attacking some of our own sites. Then I asked: "Now, how do we fix this? How do we make sure that we're not doing the same thing? We're not giving out our code. We're not doing certain things."

Discussing programming best practices was another good show vs. tell moment. For example, I talked about APIs and outlined what programmers needed to do on the front and back ends of the website, from validating data to XYZ. And then I showed them what I could do with a website that wasn't locked down: "Hey, look, I can run Fiddler.

I can do this. I can see the data. I can steal your keys. This is why you have to do certain things."

This is where the show and tells really hit home, and we did them regularly–not just for Security Sync Week.

Attack your software

Behaving like a threat actor puts software to the ultimate test. We had red teaming where we actually penetration tested our own tools, as well as using a third-party audit company. And we were pretty good at it: we had ten findings internally for every one found by external testing. When we found things, I would always show the software architect first or the group manager. A few times, we actually showed the entire team. It was a really good way to drive home the importance of application security: "Hey, we tell you that you need to do X, Y, and Z. Now let me show you why that's important."

I also sometimes did something similar when doing proof of concepts with outside vendors. My thinking was: “If the vendor isn’t doing the little things, how are they going to do the big things well?”

Build bridges with security champions

We also launched a program for security champions, which consisted of developers interested in security serving as volunteer security champions within the development scrums. They would get extra security training and a little extra money, and serve as the eyes and ears of the security program down in the weeds of a development scrum. If your process is working well, you’ll hear about things that need attention from the application security team before they become a big problem.

> From the onboarding to the fun show and tells, to hacking and showing results that illustrated the importance of application security, I tried to make sure that AppSec was baked into how our development teams thought. Having that whole full cycle goes a long way towards helping developers link the cause and effect of neglecting application security with vulnerable software. You can tell somebody to do something and they won’t do it. But seeing the actual vulnerabilities in their code and their potential impact – that really hits home.

Biography

Chris Lindsey, senior solutions architect at Mend, brings more than 33 years of development experience to his role. He is certified as a White Hat hacker and has built an application security program from the ground up for a multibillion technology company prior to joining Mend.