CERT'S PODCASTS: SECURITY FOR BUSINESS LEADERS: SHOW NOTES

The Role of the CISO in Developing More Secure Software

Key Message:CISOs must leave no room for anyone to deny that they understand what is expected of them when developing secure software.

Executive Summary

“A CISO can face the situation of being the party responsible for security, but without any real authority to control the toll gates of product releases. Implementing an SDLC gives CISOs a systematic approach for working with development [teams] to eliminate software risk. Where Quality Assurance processes ensure that software will function and perform as required, an effective SDLC brings security expertise to prevent and remove vulnerabilities. [1]”

In this podcast, Pravir Chandra, director of strategic services for Fortify Software, discusses the leadership actions necessary to get a software security program off the ground and the critical role of metrics. This podcast is based on a Fortify paper titled “CISO’s Guide to Creating and Managing the Security Development Life Cycle (SDLC),” part of Fortify’s series of CISO guides to software security assurance.

PART 1: GAIN THE AUTHORITY; BUILD THE BUSINESS CASE; START MEASURING

Have the Authority to Enforce Standards

In Fortify’s paper, they state

> Those responsible for security must have the authority to enforce standards. Clearly identify the processes and metrics that are enforceable and obtain support from management to be the gatekeeper for those processes.
>
> Document standards for security throughout the organization so that there is no doubt about what is required. Ensure that standards are preventative and proactive. Security must happen at the beginning of the process, not the end.

Effective ways to gain authority include

- demonstrate the business case for software security by
  - addressing potential impacts from software security-related risks
  - conducting assessments on critical applications
- focus on benefits such as reducing risk and cost

Have a Solid Plan

Ways to put software security on the same playing field as other technology investments include

- having a balanced plan that supports identifying activities in a methodical way
- using risk management to determine how much needs to be invested
- determining current weaknesses and identifying some easy, near-term improvements

Boil software security plans down to concrete budgets and concrete plans for improvement. These can then be compared to other project-related tasks such as feature upgrades.

To determine where best to invest

- maintain an application inventory. Many organizations do not know what they have as targets for software security practices.
- establish consistent risk rating criteria for applications--for example, is it internet facing or internal only? Ranking risks as high, medium, and low works well for starters.
- establish rules for applications in each risk rank. Apply more stringent rules to high than to medium or low.
- establish rules by lines of business. This is used as an alternative if business units operate fairly independently.

Make sure that criteria and rules are applied consistently, administered fairly, and supported by benchmarking.

Identify Metrics You Care About

In Fortify's paper, they state

> Maintain monitoring, metrics, and reporting processes that show compliance with your standards, even—and especially—if the security issue never gets fixed.

Metrics can be used to report compliance against standards, heighten awareness, and as a forcing function for issue resolution.

Poor metrics can hurt much more than they can help, and metrics can be gamed. Establish a baseline for data collection, and make sure it's consistent.

Identify metrics that reflect what you most care about, not just something that's easy to collect.

Rolling Out Metrics: Rules of Thumb

Effective ways to start collecting and reporting metrics should be based on the culture of the organization. For example, those that are more competitive could make metrics public so business units are encouraged to compete with their peers. In command and control cultures, metrics tend to get pushed down from the top.

Understand your target audience. Detailed metrics are not appropriate for upper management; you need to aggregate these and present them in business terms.

Start Simply and Start Collecting

One way to get started is to use vulnerability hotlists. Pick one or two and start tracking their number of occurrences.

Metrics that indicate whether or not the defined process is being followed can be useful, for example, are you producing design review documents and filling out the correct forms?

PART 2: SECURITY RESPONSIBILITIES THAT STICK; CISOS THAT ADD VALUE

Use Education to Set Clear Expectations

In Fortify's paper, they state

> Engage in an education campaign. Education is key to addressing security issues in all phases of the SDLC. Train software development managers on what your metrics mean. Train developers on how to fix security problems. Leave no room for anyone to deny understanding security requirements.

It is critical that someone, for example, the CISO, be designated as responsible for the software security effort.

Awareness, training, and education are the means for arming staff with the knowledge that they need. This helps ensure that they are clear about their responsibilities for developing secure software.

Build Knowledge; Measure Performance

If your senior architects understand expectations and impacts, they can help instill this knowledge in designers and developers.

Make sure managers responsible for the application understand what they are to produce. Roll out requirements to project managers, business owners and business analysts (those responsible for specifying requirements), quality assurance, and test personnel.

Make sure training requirements appear in staff goals and objectives, and then make sure performance is evaluated against these.

Eventually, make sure job descriptions reflect skills and experience in developing secure software and building secure interfaces. You can then use these as hiring criteria.

Make Sure the CISO Adds Value, Not Burden

In Fortify's paper, they state

> The CISO needs to develop a mutually effective and respectful relationship with developers. Developers should understand that the CISO is both a resource for security issues and a gate-keeper for deliverables. Listen to the problems they have with the process and tweak it to satisfy their concerns.
>
> Strive to be cooperative and supportive of other individuals and organizations as you enforce security standards. Become known as someone who facilitates solutions and makes compliance easy.

The ways that CISOs can be viewed as contributing to the team include

- demonstrating an understanding of what development is all about. This includes schedule pressures, budget pressures, and the drive to get products out the door.
- taking schedule and budget into account when they are proposing software security strategies. Imposing strict Draconian rules won't work. Fostering good working relationships and building consensus are most effective.
- positioning themselves as champions for the development team. Examples include minimizing process reengineering and the number and frequency of changes.
- integrating software security practices with existing processes and checkpoints. An example is the development and application of threat models in all life-cycle phases and confirming their use at key review milestones.
- listening to feedback and coming up with creative solutions that reflect the culture of the organization. Command and control versus lean and mean call for different approaches.

Halting Production?

In Fortify's paper, they state

> Request the authority to halt the release of any product or deliverable that does not meet security minimums. But remember that a good gatekeeper is a diplomat.

For CISOs who have this authority, it is important to strike a balance and pick your battles. If the software is critical and its release has significant business impact, an exception may be warranted.

Use the existing process and the existing review checkpoints/tollgates. Don't invent new ones.

Make sure exception processes are well understood. Track exceptions as a key metric so you know how many of these are in released products.

Managing risk is key when assessing business impact and making go/no-go decisions.

PART 3: WHEN A CISO SHOULD WALK AWAY; HANDLING OUTSOURCED SOFTWARE

When to Walk Away

In Fortify's paper, they state

> If you still end up with responsibility and no authority, walk away. Make it clear that professional ethics prevent you from trying to make the situation look better than it is. It is better that the organization face the absence of security than to let them think you are doing something about it when you cannot, despite your best efforts.

When a CISO has a good relationship with the development team, they generally figure out how to make it work. The CISO needs to frame the problem correctly, recognizing that software security is a competing priority.

As a CISO, it may make sense to walk away if you have

- the responsibility with no authority
- a management problem
- little to no grassroots support
- an uncooperative development team

Some organizations do not value security; changing the status quo is difficult.

Outsourced Software and Software Security: Be a Knowledgeable Buyer

Setting standards is necessary but often not sufficient. As a CISO, you don't have the same level of control as with an in-house development team.

Verification, testing, and code reviews become much more important; for example, make sure your supplier knows what reviews and tests you intend to conduct before accepting their software. Also, make sure to reflect software security requirements in your vendor agreements to ensure your expectations are clear.

Typically organizations are not as rigorous with open source and packaged software as they are with their own in-house developed software. If they're just getting started, they tend to focus on their own development projects.

If all of your software development is outsourced, it's challenging to gain the knowledge and experience to properly manage a vendor.

Resources

Fortify's CISO Guides to Software Security Assurance website

- CISO's Guide to
  - Application Security
  - Web 2.0 Security
  - Securing Open Source Software
  - Outsourcing
  - Commercial Off-the-Shelf Software (COTS)
  - Creating and Managing the Secure Development Life Cycle (SDLC) [1]
  - Software-as-a-Service (SaaS)
  - FISMA (Federal Information Systems Management Act)

OWASP Open Software Assurance Maturity Model

Ten Questions You’d Better Ask to be Sure Your Company’s Assets Are Secure. Fortify Software, 2008. (requires registration to download)