Designing Security Into Software-Reliant Systems

featuring Christopher Alberts as Interviewed by Suzanne Miller

Suzanne Miller: As organizations become increasingly dependent on software, the opportunity for security-related risks also increase. In today's podcast, we will be discussing the Security Engineering Risk Analysis (SERA) Framework, a model-based approach for analyzing complex security risks in software-reliant systems and systems of systems early in the lifecycle.

Welcome to the SEI Podcast Series, a production of the Carnegie Mellon Software Engineering Institute. The SEI is a federally funded research and development center sponsored by the U.S. Department of Defense (DoD) and operated by Carnegie Mellon University. A transcript of today's podcast is posted on the SEI website at sei.cmu.edu/podcasts/.

My name is Suzanne Miller. I am a principal researcher here at the SEI. Today, I am pleased to introduce you to my colleague and friend, Chris Alberts, who has been a guest on the podcast series before talking about his work with Carol Woody on security and wireless emergency alerts. As I mentioned earlier, in today's podcast, we are going to be discussing the Security Engineering Risk Analysis (SERA) Framework.

First, a little bit about our guest. Chris's research supports the development of advanced methods for managing risk and opportunity in multi-enterprise and multi-system environments. Prior to his work in this area, he co-developed the OCTAVE approach for managing information security risks and the continuous risk management methodology for managing software development project risks. He has also co-authored two books: one, *Managing Information Security Risks: The OCTAVE Approach* and the second, *Continuous Risk Management Guidebook*. Welcome, Chris, it's good to see you again.

Chris Alberts: Thanks. It's great to be here.

Suzanne: Let's start off by having you give us some insight into the current vulnerability landscape that is faced by both government and business. What types of security risks do they actually face?

Chris: I'll start with businesses first. Very often you will see businesses worried about customer information and breaches that lead to attackers gaining access to that. You can think about the Target and the Home Depot attacks in the past couple years as being examples of that. From the perspective of government organizations, it is sensitive and classified information that they are worried about. Recently, for instance, both the State Department and the White House have been attacked successfully and sensitive information was exploited in those instances.

Then, another one that hasn't occurred but is of concern, are the critical infrastructures. They are, for instance, the power grid. We are worried about whether somebody can exploit vulnerabilities and create an outage over a large geographic area. And, the impacts of that are pretty obvious.

Suzanne: OK. I remember some things about manufacturers' passwords not being reset on critical infrastructure equipment and things like that. There are a lot of different ways that these kinds of vulnerabilities can show themselves, not just in software.

Chris: Right.

Suzanne: So, we know from research that operational security vulnerabilities have three main causes: design weaknesses, implementation and coding weaknesses, and system and configuration errors, correct?

Chris: Right.

Suzanne: Which of these root causes did you focus on in this research?

Chris: Here we are looking at design weaknesses. The other two are, more or less, well researched, and there are a lot of approaches for them. So, for instance, with the implementation and coding issues, software patches are one way of addressing these. To prevent them, secure coding practices is another way…

Suzanne: Robert Seacord has done a lot of research in that area.

Chris: CERT is doing a lot of work related to secure coding practices. CERT also has a long history with operational security. For example, consider system configuration errors. When you learn about a misconfiguration, you can address that by setting the configuration correctly. So, those are addressed in those ways.

Now from a design weakness point of view, we are talking about things that really affect the requirements, the architecture, and the design of the system. So, these are fundamental flaws in the way the system is designed.

You brought up passwords earlier; authentication is a good example. If you don't really specify what the nature of the authentication in a system is, you might, for instance, have people using user IDs and passwords that are sent over the network unencrypted. Someone can steal those and then log in to the system and gain access. That's a simple example of a design weakness.

Once a system is fielded, if a weakness has something that is fundamentally affecting the architecture of the system, you may not have the opportunity to go back in a cost effective way to re-architect the system. So, you have to live with those issues.

Suzanne: Or, you have to find work-arounds that are often difficult for both the user and the people that are actually trying to maintain the configurations.

Chris: Ultimately, you are probably going to accept more risk than you really wanted when you operate the system because of that.

Suzanne: And, you didn't know that you were accepting that risk when you accepted the system. So, you thought you had a system that worked in a certain way. You find these architecture and design flaws and now you have implicitly accepted risk that you really should have explicitly known about.

Chris: So what we're trying to do is build security in by modeling risk early in the lifecycle and then explicitly articulating what that risk is and then proactively implementing controls to counteract that risk.

Suzanne: Is that different from existing approaches? It sounds like it is.

Chris: There has been a lot of work in the operational area, in systems that are operational. There's been some research. More and more research is growing in the area of early lifecycle. So, more attention being paid to it now. But, it is still largely ignored early in the lifecycle. But, for those risk assessments that do exist, they tend to be a little bit simplistic in their approach to risk.

Suzanne: So, things like mission thread workshops and architecture tradeoff analysis do look at quality attributes of the system and security being one of them. But, right now, they don't have a particular framework for saying, *Here's how we need to look at security in particular*.

It sounds like this risk analysis framework is something that could marry well with some of those other techniques that look at the architecture as a whole for risks and opportunities. But, this is an area where you could get much more specific than you typically can in those kinds of workshops in relationship to security risks. Is that a fairly good…?

Chris: Yes, that's a fair assessment. In fact, one of the models that we normally build is a mission thread. So, if you do a mission thread workshop…

Suzanne: This could be your mission thread.

Chris: This could be a follow-on piece to do the risk analysis based on that mission thread.

Suzanne: So, have you piloted this new approach with any organizations that you can tell us about?

Chris: Yes. We piloted with several organizations, all government organizations. One study I can talk you through without giving too many specifics. We did an interesting study where we took a system that was about to deployed. We did a risk analysis using information that would be available in the requirements development stage. We identified a set of controls that we felt were applicable to the system. From that set of controls, a subset of them were actually something that should have been, in our opinion, addressed in requirements. So, we created requirements for that subset of controls.

For those requirements, we compared them to the security requirements that had been developed for the system and found several gaps and missing requirements. That study kind of shows that a rigorous risk assessment might be able to identify some gaps in security...

Suzanne: That are typical kinds of gaps that people misaddress in requirements and design in the current way that our systems have been developed.

Chris: Something that I can talk about—we published a couple of reports on the wireless emergency alerts services or WEA service. We have two reports, both were published in 2014 [Click here for a link to the first report. Click here for a link to the second report].

Suzanne: Have you looked at the business context? Both of these are really government contexts. Have you looked at some of the business issues in terms of…I'm curious as to whether business architectural flaws related to security are similar or different from government flaws related to security.

Chris: We have not used this technique with industry organizations yet so…

Suzanne: But, I'll bet you'd like to.

Chris: We certainly would.

Suzanne: So if you are interested in this topic, you need to call Chris and find out if you can do that.

Chris: And more government groups and especially Department of Defense (DoD) organizations as well, we're interested in getting involved with programs that are developing systems...

Suzanne: That are early.

Chris: Early in the lifecycle.

Suzanne: I'm also assuming that early in modernization if they're—like in a sustainment—but they're doing a modernization so they're actually looking at all that requirements and design kinds of elements again.

Chris: Exactly.

Suzanne: So that's a call for action for any of you that would like to collaborate with the SEI in these kinds of areas. But what else are you doing because Chris Alberts never sits still. There's always something coming next.

So what's the next thing that you guys are thinking about? In particular, I know that you like to sort of make software program managers feel that software reliant systems will function as intended when deployed and that their cybersecurity risk will be kept within an acceptable tolerance over time. So, I know that's been an interest of yours for a while. What are you doing in that area?

Chris: One of the things that is, I think, interesting that we're starting to work on now is a software assurance framework. One of the things we are doing is identifying practices that can be applied across the lifecycle, software assurance practices. The idea, then, is that we could take this into a program and look at their practices to look for gaps, strengths, and weaknesses and identify gaps in current assurance practices, where we can address those and become a little bit stronger.

Suzanne: Chris, we do look forward to your continued work in this field and thank you very much for joining us today.

Chris: Thanks.

Suzanne: To view a technical report that Chris co-authored on this topic which provides a deeper dive into this research, please visit the SEI digital library at http://resources.sei.cmu.edu/library. In the search field, enter the name of the technical note that he recently co-authored on this topic, Introduction to the Security Engineering Risk Analysis Framework, and you'll be able to view all related resources on this topic.

This podcast is available on the SEI website at sei.cmu.edu/podcasts and on Carnegie Mellon University’s iTunes U site. As always, if you have any questions, please don’t hesitate to email us at info@sei.cmu.edu. Thank you for listening.