Security Modeling Tools

featuring Julien Delange as Interviewed by Suzanne Miller

Suzanne Miller: Welcome to the SEI Podcast Series, a production of Carnegie Mellon University's Software Engineering Institute. The SEI is a federally funded research and development center operated by Carnegie Mellon University and sponsored by the U.S. Department of Defense. Today's podcast is available on the SEI website at www.sei.cmu.edu/podcasts.

My name is Suzanne Miller. I am a principal researcher here at the SEI. Today I am pleased to have, again, Dr. Julien Delange, one of our researchers in Architecture Analysis and Design Language [AADL], among other wonderful things. Today, he is here to talk to us about security modeling tools, very important in our environment today. So, Julien, I wanted you to start by telling us sort of how did you come to be here and working in this area on security modeling?

Julien: First of all, thanks a lot for having me today. So, security. There is a lot of attention on security. We have seen a lot of bad things happening. I think one of the highlights was last year when Miller and Valasek were able to take control of a car from the couch.

Suzanne: That was very scary.

Julien: Another thing is we are starting to see a lot of issues in the medical domain with medical devices. I think people are starting to be a little bit scared about that and starting to wonder about what is going on. I mean, it is a big impact for the company's reputation. When someone discloses a vulnerability, consumers no longer buy the product because they are really scared, or they worry about it.

Suzanne: They certainly will worry about it, even if they still buy. So, what is your background that led to working in this area?

Julien: So my Ph.D. was about safety and security, so that was always a topic. When I started to see that there was a lot of attention about these topics, I started to think about developing new tools and methodologies and especially applying that for the AADL language, basically the language we are developing here.

I think it was appropriate to care about all the different issues you can have in your architecture. If you look back, most of the world tries to address security issues. They mostly care about software issues such as buffer overflows. You have issues with certification, these types of things.

Suzanne: Vulnerabilities that are in the software itself.

Julien: Exactly. But there are a lot of problems in the architecture: how you organize your resources, how you share resources among, for example, different tasks and processes. This isreally important, and, in fact, you don't have a lot of research on this.

We started to look at what is available and what are the approaches. In the UML [Unified Modeling Language] and SysML [Systems Modeling Language] community there was some work. I mean they are actually working on it. For AADL, there was nothing. Beyond that there are really few modeling tools. I think this is really something important to have tools for so that the engineer can start to model your system.

Suzanne: Before you make commitments to a particular approach.

Julien: That's correct. It's not like there are no tools, but there are really a few, and they are not used often. We wanted to make a tool that is more user-friendly and people can use.

We started to develop these tools a year ago to see how you can present your vulnerabilities; how you can see propagation of a fault.

If, for example, in your car, somebody can take control of the radio, maybe that is not a big deal. They can change the music. They can bug you and play Shakira, but it's not a major threat, depending on how you feel about it. But they can't take control of your car. So, for example, see how the vulnerability can reach your architecture from your car or not. It is important.

One or two years ago, I am sure you have seen the news. There was an individual who was going to take control of a plane from the entertainment system. When he landed, fortunately the FBI was there to arrest him. That raises the question: is it possible to do that or not? With this kind of tool you can see the impact of the vulnerability and see finally if someone can break in from the entertainment system.

Suzanne: The idea is that these kinds of tools are things that can make it easier for me to accept that risk as a consumer. But I mean, there is the consumer of the flying, but we have also got risks related to the medical devices as you said.

It is very scary to think about an automobile, which we have trusted as being this entity that is invulnerable except to things like flat tires, but it is invulnerable to security issues like we have

in our computers. But that's just really not true anymore. So, how do these models make those vulnerabilities more visible to architects and to people that are interested in verifying security?

Julien: So, we have different layers. The first thing we started to work on was just a flat model. By flat models, I mean you have to manually insert the vulnerability. You have to think, *I have a car, and I have, for example, a radio with a Bluetooth connection*. Then somebody can control the car from the Bluetooth. You have to expressly prove that. But ultimately this is not what we want. Ultimately what we want is to automate the production of this model.

So, *I have a piece of software, and then from this piece of software or model, I can generate this model. Then I see all types of vulnerabilities, which is what I will call the attack surface*.

Suzanne: The attack surface, yes.

Julien: So, all the different vulnerabilities that I have. Then I can realize a connection between the different components and see the attack impact. It is exactly what we have done in the safety community if you talk about failure mode and effects analysis. I mean, all these different methodologies.

The thing is for security, we just started to look at it. So, basically, the tool presents vulnerabilities for the component. We have a bridge to create automatically a security model from an AADL model. Just to give one example of how we can find security issues from an AADL model: if you have two tasks in your AADL model that access a shared data, if there is no concurrency protocol for the shared data, we automatically generate a security for these components.

Suzanne: So, the AADL model itself will not tell you that *Hey, Julien, you have a problem because both of these tasks can get to this data without telling the other that they are getting to the data.*

Julien: The AADL model will contain the information, but you will not see it.

The second thing it will tell you is the vulnerability propagates in the architecture and how it can impact the system.

Suzanne: That is something you can't do without a model like AADL because that is the model that gives you the insight into the architecture of the system. So, you need both pieces. You need something that models the architecture in a robust way so that you can see what all the connections are. Then you need something that is specialized to look for and analyze what are the vulnerabilities that are possible and that are actually present in this architecture. Is that correct?

Julien: That is correct. There are really simple vulnerabilities. For example, you exchange data. When you exchange data, it is critical to have encryption. Then there are different types of vulnerabilities you can have. You can basically have an attack called man-in-the-middle. A person can switch the data, modify the data. There are many different attacks we can have.

When you have such a model, it will produce these vulnerability models. Basically it will generate it. The user will be able to see the different vulnerabilities and their impacts. I think this is really what we want to do is to show the impact. For example, *I am in my seat during my flight. Can I take control of the plane? Can I basically use a vulnerability from the entertainment system, and can it propagate to the…*

Suzanne: To the navigation?

Julien: Exactly.

Suzanne: What kind of problems have you tested this kind of modeling on, or what kind of AADL models have you used this with?

Julien: We applied this technique to two-main critical systems. The first is automotive because there is a lot of attention, I think. We are coming to an era when we talk about self-driving cars.

Suzanne: We live in Pittsburgh, and Uber is actually testing self-driving cars in parts of our neighborhoods, so we have a vested interest in this.

Julien: Oh yes, definitely. It's not only Uber; I mean, you look at Tesla. I think that there is a lot of attention on this.

What is really interesting with the automotive domain is that there is not so much regulation. I mean, the regulations are not as well-defined as in the avionics domain.

We just started to work on this. I mean, there are a lot of efforts. Google, for example, has had self-driving cars for a really long time. Uber is just starting.

Many people here at Carnegie Mellon are working on this stuff. This is very complex, automotive systems need to be certified at some point, otherwise, we will see dramatic consequences. For example, I think about the investigation of the unexpected acceleration issue with the Toyota Camry. A lot of people write a lot about self-driving cars, how you can make sure they are safe to drive.

Suzanne: One of the things I know is part of this—and I don't know how this has been addressed yet, if it has yet—but in automotive, for example, there is quite a deep supply chain. You have got the manufacturer, and then you have got the people that provide the brakes, and the

people that provide the brake pads, and the people that provide the sensors for the brakes. You can go through four tiers down, and those are all different companies, and they all have different approaches to implementing things for different manufacturers. So, the bringing together all of those potential actors into the model has got to be pretty complex.

Julien: Oh, definitely.

Suzanne: Is that something that you're able to do with these—allow for multiple supply chain members to provide information so you can have some information about what is actually happening in terms of the implementation?

Julien: Yes. Remember that we started to work on AADL for the avionics domain and the avionics community. The plane is exactly…

Suzanne: Oh, yes. Multi-tier. Very deep.

Julien: Exactly. So, yes, definitely, you can use AADL to show different fields and how you decompose your system.

To come back to the previous question, we applied for the automotive domain just before the avionics domain. Last year, we decided to work also with the SAVI [System Architecture Virtual Integration] group. SAVI is a big project.

Suzanne: Right. It's a consortium of airplane manufacturers that are trying to improve safety.

Julien: Exactly. So, we have companies like Boeing, Rockwell Collins, Honeywell and so on. We apply this technique, and we apply this technique to look at two issues. One is an issue with the entertainment system. Can you take control of a plane from the entertainment system? We analyzed this technique to try to have a sense about how the vulnerability can propagate, and what it takes to basically compromise the system.

Suzanne: And then what it takes to protect from that compromise.

Julien: Correct. The second thing that we analyze is the impact of AADL ADS-B protocols. So, the ADS-B protocol is basically is used to communicate between planes and ground stations. The issue is that you have no encryption for that protocol. For example, you can inject a plane to a ground station, you can inject a plane on a plane. It's kind of an interesting issue because if you think about it, for example, you can inject a plane, so the pilots can see a lot of planes coming.

Suzanne: So, you can give false positives as well as other kinds of false information.

Julien: Yes, and you can put stress on the pilot. Of course, the pilots will say *There is no plane coming*. So it's not true for a real pilot because he can visually check, but what about the

autopilot? The computer will think there is something coming up and must react. But what about if we use the same technology for self-driving cars? It may be way more easy to do that. For example, imagine you have a self-driving car, and then you tell to the self-driving car *There is somebody that is coming in front of you; Turn Right.* What you have is you have somebody in the street, on the sidewalk, on the right. And the car will hit anybody that is on the right. With the model, we can then explore ….

Suzanne: And you explore it within a model, not on the street. So that's important.

Julien: Well, yes. We try to avoid these kinds of tests.

Suzanne: One of the things that this brings into my mind, which is not necessarily related to this exactly, but has there been any work done to take the standards for security and safety that are present in the avionics world and analyze them to say *Which of these really should be applied to automotive on the ground?* Is that something that is relevant to this work, or is it something that you know is being done?

Julien: We are not doing this as part of this project because it isn't in the scope of the project. But there are two things that…I feel that you are scared about cars right now.

Suzanne: Yes. Yes.

Julien: So, a self-driving car, there is really a lot of effort underway to summarize what needs to be done and verified for self-driving cars. You have different levels of autonomy. For example, you probably know that Tesla is not a full self-driving car. It's an assistance.

Suzanne: Assisted driving. Yes.

Julien: Exactly. And infinitely, what we want is self-driving car.

Suzanne: Although in speaking to some other researchers, I think a lot of us are of the opinion that self-driving cars are probably not going to be a complete solution until no humans are driving cars because we always inject variability into the algorithms in a way that is very difficult for the poor computers to deal with.

Julien: Probably. I don't trust myself when I'm driving a car so…

Suzanne: Case in point. So, there you go.

Julien: The other thing we are doing is to see the impact in the tradeoff between security and safety.

Suzanne: Security and safety. OK, you would think that those were pretty compatible.

Julien: Yes, basically. Some people will argue they are the same thing. Some people will say it's totally different. So, for example, if you take the example of the self-driving cars, what are we talking about? Are we talking about the security of the driver or the safety of the driver? I think both concepts can develop. This is why we are trying to see if they develop, how they develop. What is the impact of one on the other?

Suzanne: We know there are parallels. The difference between safety and security in my mind that informs this work is that idea of an attack surface and *intentional*. A lot of things that happen in safety vulnerabilities are accidental. Many of the things in security vulnerabilities are trying to prevent intentional attacks. So, there is that difference. I don't know if that is really relevant when you get to the modeling, but it may be relevant in terms of how you prioritize the kinds of factors that you are looking at.

Julien: I have a fine example. You are in your car, and you have an accident. Your car flips, and you are on the roof. What is going to happen is the door automatically opens, right? So, it's for your safety. Right now think about what happens if somebody jumps on your car, the door opens, and that person can steal your car.

Suzanne: There are some propagations that lead from one to the other. Yes.

Julien: This is what we want to investigate. This is what we are trying to do right now.

Suzanne: So, system architects that are thinking about this that are trying to improve the security and safety of their systems and who are using modeling as a way of improving that already, where should they go to get these tools and to get more information about your project?

Julien: So, a policy we have in our group is to release everything as open-source license. So, all the tools are available on our Github space. At the SEI, we have a GitHub space.

Suzanne: It's a very rich Github space. Your tools in there; lots of other tools there too.

Julien: So, the tools are called AASPE.
You go to the SEI Github website, and it is called AASPE. It's just the source code. For some people they just want to use the tool. For that, we have a nightly build. Every night we are building the tools again. It is available in OSATE, so you can build it from OSATE. [If you are using the AADL OSATE modeling platform, you can install them using the Eclipse installation mechanism and the update site at http://aadl.info/aadl/osate/experimental/. If you are not familiar with the installation mechanism, we have installation guidelines available.]

Suzanne: OK. OSATE where there are also other AADL tools, and you can learn how to use AADL.

Julien: Once you are in the platform, then you have to install what we call the experimental tools. You can do that within OSATE when you install new software Right now in OSATE it is a research project. The tool is currently available for free with the latest version of OSATE.

Suzanne: Try it and give Julien feedback when you find things that you think need to be changed.

Julien: Exactly. Also, if people want to try it, last month I gave a tutorial at Embedded Systems Week. It was a full, four-hour tutorial that covered AADL modeling, safety modeling, and security modeling. In fact, you can find the tutorial easily on the Github website. We have slides. We have instructions that explain how you download the tool and use the tool. We have models to use. People don't have to know AADL. They can just see how it works.

Suzanne: Excellent. So, you are making it much more accessible to even people, and you're going to get people interested in ADL because once they see how cool the models are they're going to want to do AADL, yes?

Julien: Of course.

Suzanne: I do want to thank you for joining us today and talking about this. The more we can do in modeling, the less we have to do on the ground or in the air. That really makes it safer for everyone, I think. So, I am really pleased to see this direction.

You do have a blog post that is on this work. That is another resource for our viewers. Those are available at insights.sei.cmu.edu. That's where all of our blog posts are. So, I invite you to go there, find Julien's blog. His last name is spelled Delange, D-E-L-A-N-G-E. Not everyone has the French accent to go along with it, so I spell it out for them.

We will include resources to Github and other OSATE resources we mentioned; those will be included in this transcript and the podcast itself will be available, as I said, at http://www.sei.cmu.edu/podcasts.

It is also available on the Carnegie Mellon iTunes U site. Thank you all for watching, and thank you, Julien, for talking to us today.

As always, we include links to these resources in our podcast transcripts.

This podcast is available on the SEI website at sei.cmu.edu/podcasts. It is also available on the Carnegie Mellon University's iTunes U site. Thank you very much for viewing today.

As always, if you have any questions, please do not hesitate to email us at info@sei.cmu.edu. Thank you.