Moving Target Defense

featuring Andrew Mellinger 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. This podcast and the series are available on http://www.sei.cmu.edu/podcasts.

My name is Suzanne Miller, and I am very pleased to welcome today Andrew Mellinger, one of my colleagues and friends. Today we are here to talk about moving target defense. I will let him talk a little bit about what is your background, what brought you to this work, and then we will get into what is this all about.

Andrew Mellinger: Sure. In general, I am kind of a software engineer. I do the full spectrum. I really focus on software architecture but also on adaptive and dynamic systems. That is an obvious fit for when we talk about moving target defenses, which we are now talking about. We term them as dynamic network defense, so obviously that has led me into that area.

Suzanne: Why is dynamic defense an important topic?

Andrew: When most people think about defense, whether it be a network or a physical entity, they think about a static set of defenses. We are used to thinking about *How do I defend an organization?* Imagine a brick wall, a strong door, a gate, or something like that. All those defenses, what they evoke is this kind of big monolithic, static set of walls. Within enterprise networks, what we find is that that gives a lot of opportunity to our attackers to understand what we do.

Suzanne: Because they can see the wall.

Andrew: Absolutely, absolutely. It gives them time because we do not change it over time. While that is a natural thing to want to do when building your defense and maintaining—and not just your defense but your entire enterprise—what we try to do is say, *if we move those things around, how could we make it harder for the attacker to understand our environment*? That is the fundamental premise. It is as simple as that.

Suzanne: How do you go about making a defense dynamic instead of static?

Andrew: In most cases, you actually can't take an existing defense and make it dynamic. What you end up having to do is redesign or re-envision your systems to have dynamic attributes.

Imagine your computer, which has an IP address. What if that IP address was changing over time? You were not at the same address, you were somewhere else. When the adversary did their initial reconnaissance, they would find you at a particular address. When they went to come back and compromise your machine, it could be somewhere else.

Suzanne: That sounds like a productive way of foiling that kind of attack. Are there other specific kinds of attack that dynamic defense is particularly geared to defending against?

Andrew: Actually, they would cover all of the spectrum; so anywhere from a host, any action with a host, whether it be memory layout, instruction layout; all the way up through networking stacks and protocols and network routes; all the way up into how we actually work with our computers. Most organizations have a policy where you have to reset your password.

Suzanne: We know that one well here.

Andrew: What if you had to do that on a weekly basis instead of a monthly basis?

Suzanne: As a user, I give you a heavy sigh because it is problematic from the user interface viewpoint of remembering all those things, but you have a solution for that I will bet.

Andrew: Well, actually we do not yet.

Suzanne: Darn.

Andrew: That is one of the hard problems. What you brought up was, as we change that password, as I force you to change that password to make the defense better—if the adversary had been cracking it, and now you have a different one, they would have to start over. How do I make it so that I can have a dynamic defense, but it does not cost the defender a lot of money?

Suzanne: So patterns that you have, ways of manipulating in unpredictable ways is one of the strategies you might think about for something like that.

Andrew: Right. Also how do I make that information. So how do you remember your password if you change it all the time? How do you?

Suzanne: I am not telling on video.

Andrew: We do not have a good solution for that yet. Well, imagine if you are a systems administrator, and you have lots of password changing or network layouts. We have to build a set of tools too that enable us to understand that dynamic environment.

Suzanne: So is that – you have got a platform that you are working on to actually help to validate and vet new technologies that are in this arena trying to help with these kinds of problems?

Andrew: Yes.

Suzanne: Tell us about that.

Andrew: The platform—we call it middleware because that is basically what it is—it deals with sending around messages and information between all of the various machines that are involved in your enterprise. Its job is to kind of mediate between that dynamic change and the static environment. So, it will help a machine move and be able to send back information and say, *I am still the same machine.* Now obviously, you are going to say *Well, if you are doing that, couldn't the adversary tap into tha*t? Well…

Suzanne: I was going to say that.

Andrew: Right, you were probably imaging that. That is a common question we get. So the answer is, that is the obvious point of attack then. Attackers are going to come after us, because we control that environment instead of going after a lot of other things.

We know that, we have to work especially hard at designing in security and imagining those scenarios are going to be like. But, since we are starting from the beginning, we can build security in from the ground up.

Suzanne: Build Security In is one of our big themes in all of our security work at the SEI. You mentioned the difficulty of moving from a static to a dynamic environment. Do you have any sense of what is the proportion of organizations, enterprises that are moving to dynamic defense at this point? Or, is it just in that *too-hard-of-a-change* kind of category?

Andrew: We see—what they call moving target defense, before a moving target, trying to envision the idea of somebody running across a field—is being implemented by a variety of companies out there, but typically they are point solutions. They might change how a webpage is laid out internally to make it harder for somebody to attack.

They might do things like change your password, change your network connection. What we do not see is people doing that systematically across the enterprise. We do not see them taking one vendor's defense, another vendor's defense, and trying them together.

Suzanne: Which adds more complexity, and more complexity in the defense means it takes more complexity to attack. So at least some proportion of attackers, are going to give up and say, *forget it. You are too hard, I will go to an easy one*.

Andrew: We hope so. Certainly, even if they do not give up, that time that it takes them to penetrate allows our defenders to find them. We do not have the assumption that we are going to stop the attacks because we are never going to stop all the attacks.

Suzanne: It is the beginning of the school year, and the number of things [suspicious emails] I have been sending to our suspicious email addresses [IT security] is skyrocketing even with all the filters we have. It is like, *They are back*.

Andrew: Exactly, and the more we can make that obvious to the defender, the faster they can get in there and make better decisions.

Suzanne: OK. So, what kinds of things have you found in using this platform? What are some of your research results in terms of things that work, things that do not work? You may not want things do not work.

Andrew: We are still very early in that. Our research in particular is not so much on which defenses actually work together but really on the complexities and the challenges of making them work together. As I mentioned before, one of the key challenges is when a defense is actually changing things, how do we watch that?

Suzanne: How do you gain the insight in terms of what the attacker's doing?

Andrew: Actually, no. *How do we use that change so it does not confuse the defender*? If you are a system administrator, and your machine is moving around, you have lost control. How can you make them move around from the perspective of the attacker, not the perspective of the defender?

Most vendors don't think about that. They say, *we have got this black box solution, you stick it in line, and all is good to go*. Then the forensics guy has to come in after the fact and say, *well, this thing did this stuff, and I do not know what it did. I cannot really figure out how it slowed the attack, or how the attacker got through, or anything else*. They are not that easy to use from the forensics perspective.

Suzanne: This is a team sport. Obviously, there are lots of people that have ideas about many of these aspects, which attributes are more amenable to becoming dynamic, et cetera. Tell us a little bit about the collaborators and the collaborations that you have used to make this happen.

Andrew: Folks on campus in the Institute for Software Research, they have been looking at self-adaptive systems now for probably going on 15 years. That is part of an effort called autonomic systems that came out of IBM back in the early 2000s. They have been doing research into the broad ideas of self-adaptive systems.

So, ideas like *how would I increase the performance of things*? They call it *self-optimization*. *How do I repair the system*, so self-healing? *How do I configure the system automatically so I do not need to do that*, self-configuration? And, the fourth part of that is self-protection. We really have been focusing that sort of adaptive research community on the self-protection part.

Suzanne: That sounds exciting, and it is always good to work with ISR. They are a wonderful group of folks to work with.

Andrew: They have a huge amount of traction in that space. Being able to say, *OK, take those ideas you did for performance, right? How you talk about latency and bandwidth and adaptability and lead times, and all these sort of things from a performance perspective, now adjust those for protection.*

Suzanne: OK. That makes sense.

Andrew: It is not just, we are cooking up something…

Suzanne: It is not pure invention, it is adaptation.

Andrew: It is self-adaptation.

Suzanne: Where are you now, and where are you headed in the next year or so with this work?

Andrew: We are at the point now that we have our middleware platform working. We use this thing called the MAPE-K Loop that is the autonomic system. It is kind of a monitor/analyze/plan/execute, we have this loop.

We have implemented a loop, and we are actually now building out the adapters that run some candidate systems. So we brought these systems—well, we are bringing them in—and wire them up. And, we are starting to pit them against each other, and see *if we turn on both, what happens? Will they work together, or will they not?*

If they will not work together, how do you build this self-adaptive systems loop to recognize that? And be able to say, *OK, well, under these conditions, I should be using that one*. Then 20 minutes later, *under these conditions, I should be using that one*. How can it change those over?

Suzanne: …using them together is not productive in this case.

Andrew: In some of these, and some of them they are. How do you have these advanced enterprises be able to bring up more than just a handful of complex interactions across the entire, think thousands, tens of thousands of machines?

Suzanne: When you get that problem about the changing the password every week so that the user can still remember it, when you get that one figured out, I want to play because I really need that one.

Andrew, I want to thank you for sharing your work with us. Self-protection, we all need that. There is this tension between user usability and protection, and this dynamic kind of idea adds a little more complexity in, but it does give you more protection. I can see exactly where that is going, so we have got to find some of those solutions, and I know you guys are working on that, so thank you.

I do want to say that you were highlighted, your work was highlighted, in the SEI's Year in Review for 2015. That is our annual publication that talks about work across the institute. So that is one of the places the people can find your work, and a PDF of that publication is available at sei.cmu.edu/yearinreview, that is "year in review" all together.

The SEI Year in Review highlights work from our past fiscal year and showcases the support that we provide the DoD and other organizations in acquiring, developing, and deploying trustworthy software-enabled capabilities.

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.