Threat Modeling: Protecting Our Nation's Software-Intensive Systems

Featuring Natasha Shevchenko and Alex Vesey as Interviewed by Tim Chick

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

Tim Chick: Hi, my name is Tim Chick. I am a tech manager here at the SEI. We are here to talk about building software intensive systems from a cybersecurity perspective. What we are thinking about, and what we are concerned about is, *How does someone attack my system? How do I defend against it?* One way of doing that from an architectural system engineering perspective is to think about threat modeling. Threat models are really important because they guide requirements, system design, operational choices to really identify those threats and then ultimately create mitigations to it. With me today I have two of my colleagues, Natasha Shevchenko and Alex Vesey. Welcome. How we start all of our podcast is, *Tell us about yourselves and what brought you to the SEI and why do you why do you like working here?* Alex, I will start with you.

Alex: Sure. Out of college, I went to work for a defense contractor for about six years. I really like the defense contracting space. We get to work on hard problems. When it comes to hard problems in the realm of software

engineering, I think the SEI gets to work on some of the more difficult ones. That is why I ended up here.

Tim: Yes. Great. Natasha?

Natasha: I, first time, got into SEI when I was a master's degree student in CMU, Carnegie Mellon University, and I took some classes here. I met a lot of very interesting people. At one point, I was looking for a job. SEI was one of my first places to start. I really like working at the SEI. One of the reasons is because it allowed me to find interest in working on very hard problems to solve. Before coming to SEI, I worked in the banking industry, and railroad, and communication. I bring an interesting experience here.

Tim: In lots of different domains.

Natasha: Like different domains. Correct. It helped to apply all of my previous knowledge for a couple of decades.

Tim: We are glad we were your first choice. Alex, you recently wrote a blog about threat modeling and model-based systems engineering or MBSE. Can you tell us what that is about and where it came from.

Alex: Sure. Yes, there has been a big push, especially across the government but in industry as well, to start doing digital engineering, which is moving away from the traditional documents-based approach and couching more of the system design and information in digital models. A subset of that is model-based systems engineering [MBSE], which is focused on managing the development of the system throughout its lifecycle. Marrying that with, as you mentioned before, the threat modeling, especially early in the system's development lifecycle, I think has major benefits for being able to build secure systems and systems that are secure by design.

Tim: Right. There is a great plug for that. We do have a conference event coming up called Secure by Design. Back to your point there, two points there. One is transitioning from a document-based to a model-based system engineering. What I like about that is you say it once but reference it often. I get different views of the data based on my engineering discipline and based on my what I need out of the requirements or the design, but it is all one set of truths. With document-based, I never know which version is true. You updated this version of that document, but those changes haven't been captured or the related changes have not captured other documents in the other document. You always had this synching problem. I have always really

liked the fact that MBSE and the digital engineering approach, really solves that.
Natasha, you have done this for a while, model-based systems engineering as well as threat modeling. You actually have a couple of older blogs from several years ago out. What is the current state of threat modeling in terms of system engineering?

Natasha: Threat modeling still lives mostly on the low level of design. When design of a system already exists, and then engineers start to analyze it from a security point of view and try to model there. There are attempts to move the threat modeling to the left so-called to start to work on that earlier. There are a couple of articles [See here and here] about the inherited threats, which is coming from a business level that affect the system. You can't move it. But the threat modeling as a methodology is still pretty much more art than engineering. There is no one simple standard. There are a couple of methods, but they mostly rely on engineers and their experience and yes, expert judgment. Again, the level of quality of your analysis like that depends on this expert level.

Tim: Right. There are now various attack frameworks, and there are other standards and examples of different methodologies that have been found in the wild in terms of attacking systems. You really have to start studying those things and being aware of the different ways in which your system can be attacked, that cyber analyst type of role comes into play.

Natasha: They are mostly standalone. It means that the ATT@CK, it is one of the MITRE taxonomies on tactics of threats and attacks. There is another taxonomy that it is CAPEC. There is a high level. Again, MITRE produced it. There is a connection, like secondary, but it exists in the form of links on the website or at best as a spreadsheet. They are not presenting the coherent, one space on threats, threat types. or methodology. It is a treasure trove for analysts, but it is hard to use on a high level of architecture, for example, because they concentrated on specific tactics and specific implementation details of a system. It is very hard to think about cybersecurity and specifically threats when you are just planning your system or planning your infrastructure, and you don't know exact details. But, sometimes if you miss this, and you start to think about something in detailed design, it is a little bit too late to fix this problem.

Tim: It is more expensive.

Natasha: It is much more expensive and adds to the lifecycle.

Tim: It takes more time and all those things.

Natasha: Exactly. There is a gap in tools to bring these different frameworks together and make it available to architects, to software architects, to maybe even enterprise or domain-level architects, so they can look in these taxonomies of threats and think how they apply to their system if they have similar problems, similar vulnerabilities just when they plan their systems.

Tim: Although I still find it encouraging that those resources exist. While it is still hard to take those and abstract out to the left, the fact that we are now thinking about it is still kind of revolutionary. Because five, ten years ago, cybersecurity is really just very compliance-based, right? *Here. Check my box. Do these things*. It was also very much to the right. If I find like engineers versus cyber experts, they actually have very different vocabularies because they came from very different places. Software developers came from the engineering domain and cyber analysts really came from system admin/ operational domains. Getting that that vocabulary and start transitioning into engineering language I think is just huge because what I really want and what I really need is software developers to think and understand that their decisions and their limitations have consequences. But the sooner they know that, the sooner they start thinking about that, the sooner you actually have a secure system by design, *Not hey, I checked the box. I built a moat around my system. I hope I can defend it*. It is a very different mentality.

Natasha: Yes. If we can bring it even earlier when enterprise architects think about the system, they should be able to think about the level of a big piece of infrastructure. It is not just like computers and nets and other subnets. It is computer centers. It is data centers. It is the security enclaves, that can and…

Tim: Or large weapons systems. A weapons systems is a system that also has lots of support systems and supports. So it is this enterprise. It is not just a single element.

Natasha: Yes. Most of the time the enterprise architect, they know they need to think about security, but there is no methodology with ready ontology to express these concerns. If you are using specifically MBSE, which is covering the whole lifecycle, you have to reach those gaps, to go through the actual artificial borders that extend between the level of architecture. So all architects from enterprise to software can talk the same language and understand each other and even pass to engineers the solutions they can implement. This traceability from the very early stages of design of a system

to implementation of a system, this is the gap we are trying to close in the work we are doing with Alex and his blog post [Stop Imagining Threats, Start Mitigating Them: A Practical Guide to Threat Modeling] is actually talking about how to address threats of your system and then mapping it to the existing standards. It will improve your standing and prove that what you are doing makes sense. It is not just in your head. Industry supports you and industries like with the standards show you there is possible mitigation for that, or there are some security controls that you will be compliant with if you close this gap, if you mitigate this specific threat.

Tim: Alex, you are relatively new to some of these tools and techniques. You are very experienced as a developer. You moved up to more systems engineering thinking. If the audience comes and says, *Well, how do I get started down this path?* What was your journey that got you to where you are where you are now writing blogs about threat modeling and MBSE.

Alex: Sure. I think it follows the progression that you had mentioned where a lot of cyber analysts come from an operations or a developer background. You start with the very detailed, maybe more narrow view of mitigating particular threats in, say, an operational system. There you are again looking at very specific, a solution architecture or something that has been implemented. If you start there with understanding how a particular threat is mitigated on a particular system, I think then you can start to transition your thinking to maybe more broadly, how does that class of threats get mitigated? Ultimately, as a software developer, what sort of patterns or architecture that I implement that might be able to eliminate, say a particular class of threats completely from my system. I would say that is my idea of how you might go from, say like a software developer, someone who is maybe not super threat conscious or isn't thinking about maybe the higher-level design implications of a particular threat to having that broader view I suppose.

Tim: I think most developers don't really understand or realize that they are implementing an architecture. A lot of times they inherit an architecture. For example, microservices is a well-known, well-used modern technology. But that is actually an architectural pattern. That is not necessarily the right pattern for every solution. Sometimes developers just blindly use it because that is what everyone else is using. When we talk about model-based systems engineering and taking it up a couple levels of abstractions, *What am I actually trying to build so I can pick the right patterns to find the right solution*. Once you start realizing that, that is probably when you started over. There more to this, right?

Alex: Yes. Natasha mentioned earlier there are inherited threats that you just get because you pick, say, a certain architecture or even maybe just because you are engaged in a certain type of business. That is getting pretty high level to more of almost like a business strategy kind of level. To your point about like the microservice architecture, there are certain threats resident in that architecture that just wouldn't be there if you deployed a monolithic application.

Tim: Which is beneficial to both. If I pick a pattern, I should be able to figure out what are the inherited risks or cyber threats to which I need to mitigate. There are benefits also from taking a pattern. There is research and literature out there that you use to try to find help guide you locate. *I use this pattern, so I need to worry about these things.*

Alex: That is also where getting systems engineers and architects to talk to each other can also help promote cost savings and mitigate risk at the same time because if you choose an architectural pattern that has fewer, inherited threats for your particular domain, then all of a sudden you don't have to implement a bunch of security controls on the back end because they are just simply not applicable.

Tim: Right. Forcing me to do the wrong solution to meet your need always costs more.

Alex: Right. That is where you end up kind of implementing those security patches on the back end where it costs more and can limit the system capabilities.

Tim: Natasha, in terms of what is the current state, it really is very immature in terms of threat modeling and things of that nature. Part of your work is unified architecture framework standards. I have worked with you in the past where we try to do threat modeling using UAF , and we really found it was missing some core elements. Can you talk about some of those core elements that the current UAF standard is missing that we are working to try to actually improve to make it more useful from a cybersecurity perspective?

Natasha: Yes, UAF is the architecture framework, Unified Architecture Framework (UAF), and language that accompanies it, is built to describe the correct way of architecture. So how we need to build our system so it will perform according to requirements and so on and so forth. What is missing is the vocabulary to describe the fault state of a system. *What happens if my*

system is misbehaving? What happens some actor will perform something that puts my system in a fault state? There is no lingua in UAF to describe it. One of the first questions, the main question, when you do threat modeling and you ask, *What can go wrong with my system? What can go wrong in this specific component or here when these two components interact with each other?* There is no way to describe it. You need to have an extension to the standard to describe it. That is actually what we did. We created a custom profile that extends UAF that allows you to describe a situation where my system can break in some way, can produce a fault result or not produce a result at all, for example, which will be a failure of a system itself. This allows us to start this kind of analysis, to describe the situation, to describe the actor/potential actor who can do that. It allowed it to connect it to a place in architecture where this fault can happen. *Is it a component? Is it an interaction? It is the process, like a business process, that something has gone wrong?* It is like a starting point for analysis. This is the first step, finding the threats. Finding who can do that. What are the facts of this threat? What happened and or why we care about this specific threat. Then we can start thinking about next step. OK, we have this vulnerability in the system, how can we fix it? The next step is mitigation. UAF gives you the language to talk about mitigations.

Tim: You have relate it to the actual threat that you are trying to mitigate.

Natasha: Yes, but otherwise they have a risk element like a risk, but it is not enough. It is not enough. Risk is undeveloped, in some way, especially for cybersecurity threat analysis, it is an undeveloped element. So we extended it to cover specific needs for cybersecurity analysis, specifically threat analysis.

Tim: Alex, you keep using the word threat. It is as if you have different imagination of the what that might mean. In your mind what is a threat from a system engineering architectural perspective?

Alex: In the model the threat as we have defined it, really a scenario that incorporates a couple of key elements that include an actor. So who is actually going to be carrying out the attack? The effect that that threat is going to have? What is going to go wrong with your system? The objective, that someone might be trying to achieve.

Tim: Why would they even want to do it?

Alex: Right. Then, crucially, a separate-but-related element is the attack. The specific way in which a threat actor would achieve their effect and their

objective on the system. The threat is kind of the combination of all of those, because you have to have both an actor that is capable of doing a particular act to cause your fault state as well as the vulnerability or the underlying weakness in your system that they can exploit. It is a combination those three elements.

Tim: So we use the word *actor*. A lot of people in cyber they think about a persistent threat or a nation threat. That threat could actually be internal. It could be your developer, right? And their intent could just be laziness. It could be, *I didn't do a code review. I didn't run my static code analysis* right, so I have this weakness. Right? I just want for the audience to understand, like an actor can be anyone, depending on what the system is that you're that you're defining and the type of threats that you're, that you're perceived. The other part is understanding what the impact of that threat is. That is really important if you're going to prioritize. You can't mitigate everything. Right. If I want to mitigate it, to truly mitigate 100 percent of a system, I basically turn it off. Right. And so it becomes unusable. There is always a balance between security mitigations and like usability.

Alex: Right. And so that multiplicity of having multiple different actors, like an insider threat or an advanced persistent threat that could cause a problem in your system is actually, I think, one of the unique things about how we have integrated our method into UAF. A lot of other threat modeling methods will incorporate things like attack or killchains. And those are good because it shows you like the end to end, like how an attacker goes from accessing your system all the way through, causing an issue. However, in some instances, those attack trees or chains can be somewhat limited because it requires you to walk the entire sequence, right?

Tim: They also assume you are at the solution base at this point. You kind of know what technology you are using. You are at the implementation phase for that type of analysis usually.

Alex: But each step of the way, every way that a threat say leverages a particular vulnerability, it could be different actors along the way that cause or reveal those different vulnerabilities. Being able to think about the individual pieces of the chain and not have to lay it all out to start I think is somewhat helpful because it doesn't matter how a system got into that fault state, just that it got into that fault state. That is what we are trying to avoid.

Tim: Natasha, I talked to you about the secure software design, which is more focused on software developers and how can they start thinking about

security, like using good engineering techniques to do that. You have another event coming up MBSE in Practice, also I think in August. What is that focus like? How is that kind of scratching the itch of these things that we talked about?

Natasha: MBSE popped up to the top in industry in the last five, six years approximately, and it combined the multiple methodologies, multidisciplinary thing. And learning MBSE methods and practices is pretty sharp, a pretty steep learning curve. Very often people who see it for the first time, they don't know where to start. How it can be helpful for them because [everything happens] so fast? Different tools are involved, modeling languages are involved. Some, domain specific extensions to the languages, the methodology involved. A lot of theoretical work going around like, books and articles and so on. The engineers and practitioners actually have the problem that they need something concrete to take in and use it right now.

Tim: To get beyond the theoretical and how do you do this in practice? *What are best things you learned? What are the practical tools and techniques and tricks of the trade?*

Natasha: And it should not be vendor specific. Actually, it is not necessary to be domain specific. Something for, like automotive company or defense something. There is a common thing. There is a common practice that MBSE practitioners can use right now to solve very specific problems they have in their companies. I hope that we get a variety of speakers that will cover all aspects of adopting MBSE, using MBSE, extending the practices, and bringing the case studies, how they use it. I know we will have an absolutely awesome speaker on site here.

Tim: It really trying to create a forum to bring people who are actually practicing it. *Let's stop talking about theoretical stuff. How do you actually do it? Let's come together, and share those best practices.*

Natasha: Yes. They use it not only on examples or prototypes, but they use it on real systems, and it worked. They bring this experience to share with fellow MBSE practitioners.

Tim: I look forward to attending that event. What's next? What are you guys working on that might bring you back for a future podcast? Alex?

Alex: Yes, in this blog we talked a lot about integrating CAPEC [MITRE's Common Attack Pattern Enumerations and Classifications (CAPEC)], however,

I think there is a lot more work to be done in integrating other frameworks, other taxonomies to give cyber analysts, threat modelers a larger library of ready to go threats and mitigations so that you can take the most advantage of MBSE and the MBSE tools to rapidly model and mitigate threats across not just software in general but even some of the more specific domains.

Tim: I look forward to it. Natasha, what is going to bring you back here?

Natasha: I think what would be interesting to talk about is actually how to model comprehensive cybersecurity architecture on the high level and talk about specific techniques, how the architects can use MBSE, can use a specific UAF, to model not only the correct the final architecture of the system, but start to think about the fault states, so they can architect in the solutions for the security for this system. Thinking only about happy cases won't help. You need to think about the worst cases, unexpected cases, how your system can be abused.

Tim: That is the harder path, right? The happy path is always easy. That is the ideal path. *I can't imagine why anyone would push that button, or do it that way*. Well, they will because they can.

Natasha: Yes, and these questions can be should be answered as early as possible.

Tim: Mitigated and most cost effective and resource way.

Natasha: Yes, I would. It is a known thing that the worst problems are introduced in requirements and architectures stage. They are more expensive to fix.

Tim: They are the ones you don't find here in the operational phase. You have gone through the entire engineering lifecycle, and the money is gone, the schedule is gone, but you have this defect.

Natasha: Yes, too often it is only when the system is alive that you find this error.

Tim: I find that synonymous really with what this technique is all about.

Thank you for joining us. This is a great talk. To those who joined us, thank you for listening. We will provide links to any of the material that we referenced today and the transcripts so you can find that material. The SEI

Podcast Series is available in all the places you can find podcasts: Apple Podcasts, SoundCloud, Spotify, and the SEI's YouTube channel.

Thanks for joining us. This episode is available where you download podcasts, including SoundCloud, TuneIn radio, and Apple podcasts. It is also available on the SEI website at sei.cmu.edu/podcasts and the SEI's YouTube channel. This copyrighted work is made available through the Software Engineering Institute, a federally funded research and development center sponsored by the U.S. Department of Defense. For more information about the SEI and this work, please visit www.sei.cmu.edu. As always, if you have any questions, please don't hesitate to e-mail us at info@sei.cmu.edu. Thank you.