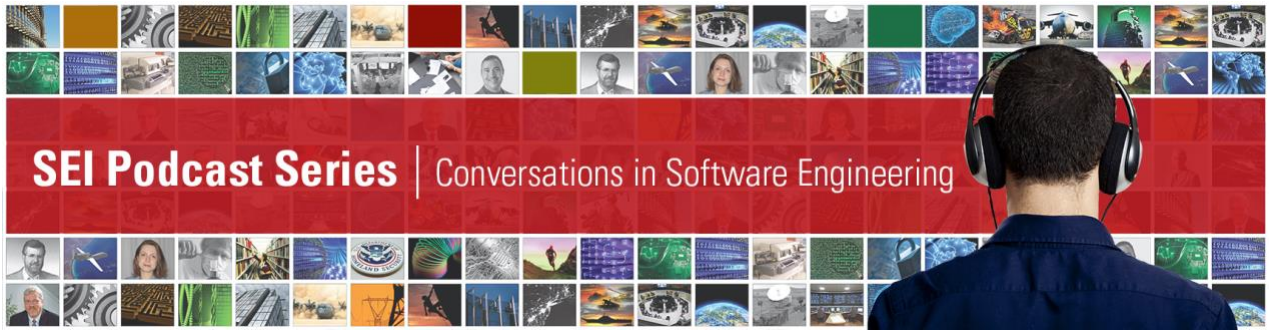

Managing Vulnerabilities in Machine Learning and Artificial Intelligence Systems

featuring Allen Householder, Jonathan Spring, and Nathan VanHoudnos

---

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.

Jonathan Spring: Hi, everyone. Welcome to the SEI Podcast Series. My name is Jono Spring, and I am a senior vulnerability researcher here at the Software Engineering Institute's CERT Division. Today I am pleased to welcome Allen Householder and Nathan VanHoudnos in discussing the management of vulnerabilities in machine learning and artificial intelligence systems. Welcome to you both.

Nathan VanHoudnos: Thanks, Jono.

Jonathan: Let's start off and talk a little bit about the work that we each do here. Allen, do you want to start and talk about what your work is?

Allen Householder: Sure. I too am a vulnerability analyst within the CERT Coordination Center at the Software Engineering Institute. I have been doing this for a number of years. I started off in incident analysis and have done some malware-analysis stuff along the way. The common thread that runs through my stuff tends to be more modeling and math and simulation things surrounding vulnerability management, vulnerability analysis. I have done some work on fuzzing systems. I have done some work with coordinated vulnerability disclosure. But generally applying different ways of modeling the processes and things that can go wrong on the software vulnerability side of the world.

Jonathan: Great, thanks. Nathan, how about you?

Nathan VanHoudnos: I am a senior machine learning research scientist here at the SEI. I am essentially running the adversarial machine learning portfolio. Adversarial machine learning is this idea that we have machine learning systems that do different things for us, and we also want to often run them in situations that have adversaries. So, for example, an adversary might make your machine learning system do the wrong thing, it might make it learn the wrong thing, it might make it reveal the wrong thing about your training data. Most of my work here is about understanding how we do machine learning well and understanding how we can make it secure and robust.

Jonathan: Great, thanks. Over the last three years, we have seen a lot of examples where real-world machine learning systems have been tricked in one of the three ways that you mentioned, Nathan. Google, Amazon, Microsoft, Tesla all have seen examples of this. What does the current threat landscape look like for machine learning systems, and how does that intersect with our work in vulnerability coordination?

Nathan: Generally what happens is that you have a bunch of academic researchers that try and take these different machine learning models or machine learning systems and play with them. You mentioned the Tesla attack. That was a lab out of Tencent looking at what a self-driving car would do and figuring out how they could in fact cause it to change lanes by only modifying a little bit of the road, specifically putting some small stickers to make it think that it needed to change lanes. In terms of thinking about the normal threat landscape that we would think about in the cyber domain, what we mainly see is proof-of-concept stuff in the machine-learning systems. We haven't really seen a lot of very specific robust attacks to machine learning systems at this point, but that doesn't mean that they are not coming down the pike.

Allen: I think on the traditional vulnerability-coordination side of things, what we have seen is we have had two vulnerability notes that CERT has put out in the past couple years about machine learning-related vulnerabilities. They are actually examples of two very different abstractions of vulnerability. One being a vulnerability in the antivirus system, where it was trivially able to bypass the trained model because you just appended a little bit of text to the end of the file, and the executable could still get through, but the detection would fail. That was an example of a specific instance, specific product, kind of a traditional vulnerability, it was just a novel method for how that vulnerability was exploited. The second one that we put out was about any system using gradient descent being vulnerable to a particular kind of attack, and essentially it comes down to…and some of the impetus for the example from this paper, are this idea that the problem might actually lie in just the way the system was built. It is not necessarily that it is a code bug. It is not a thing that you can go patch easily. It is just, if you use this technology, you will have vulnerabilities or have problems that could be vulnerabilities like this.

Jono: Nathan, you mentioned that the attacks are not robust yet, does that mean that they are hard to find? Are there relatively few known attacks, or is it pretty easy to find an attack on a machine learning system?

Nathan: It depends on how much access your attacker has to the system. For example, if your attacker has a copy of your trained model file, like actually can load the neural network, for example, that you use in memory, then they can design something that will fool it very simply. If your attacker does not have access to your model, their job becomes harder. Other than sort of adversary knowledge, another issue is what kind of control they have over the input to that model. If they can actually modify, say, the images that are coming in, and directly modify specific pixel values, then they can totally just arbitrarily control what the system does because they can control its inputs exactly. If they instead have to do something like put physical objects on the ground, as in the case of the Tesla example, that is a lot harder, and making those attacks robust is much more difficult. What I mean specifically by *robust* there is that, we have a self-driving car designed by Tesla, so we have an attack against Tesla. What about a different self-driving car made by a different company? That attack against Tesla may not work against that other self-driving car. That is what I meant by *robust* in the attacks.

Jono: Interesting. OK, thanks. We talked a little bit about this stuff in a paper at the New Security Paradigms Workshop last October. Allen mentioned the paper. The thought experiment that we walked through there was, so there is a problem with a Tesla, the machine learning system. Does something like that get what the vulnerability-management community understands as a vulnerability identifier, a Common Vulnerability Enumeration ID? CVD ID? I know we used a different example in the paper, but everyone could go read that if they want.

Nathan: Please do in fact.

Jono: Yes. But for the flaw in the Tesla where you can make it change lanes, or the drone can project the stop sign on a tree and cause the Tesla to stop. Allen, do you think that those are things that should be given CVE IDs to facilitate communication to, say, the owners of Teslas?

Allen: Whether or not it gets a CVE ID might be distinct from whether or not it should be a vulnerability and treated like a vulnerability. To be clear here, CVE IDs are an identifier that are assigned. MITRE Corporation actually operates the CVE list, and it is intended to be a dictionary of identifiers and then the definition, which is basically a description of the product and the vulnerability and something that uniquely identifies it from other vulnerabilities, potentially in the same product, even if they are potentially similar vulnerabilities. But not all vulnerabilities get CVE IDs. So, for example, a vulnerability in a website may not get a CVE ID because there is really only one instance of that website, and once it is fixed, it is fixed, and there is nothing for the users to do. So, if it requires the users to take some action to patch, then having IDs so that

you can coordinate those actions can be useful. If there is nothing for the user to do, if we are just going to patch automatically, and I am not familiar enough with Tesla in particular to know, do they have an auto-deployment system that they can just push patches out? If they can ,and if it is totally under the control of the vendor, the supplier of the patch, maybe it does not need the ID.

In the paper, we go into a lot of gory details of *MITRE has put out a list of criteria for how to know if you should assign a CVE ID, how many IDs should you assign given the description of the bug and everything*? I do not want to belabor that because it was pedantic enough going through it the first time to write the paper. It will bore most of the folks listening to me right now. But the end result there is, given those rules, you may or may not come out to that answer of assigning an ID or not. That does not necessarily mean that it should not be a vulnerability, and that you should not treat it the way we treat other software vulnerabilities.

Jono: Yes. So, Nathan, with this sort of a situation, who currently talks about these sorts of problems. What is the community that is currently talking about them and how does it intersect with the community of people that use CVE IDs to talk about problems?

Nathan: That is a really interesting question. Adversarial machine learning [AML] as a field is this academic discipline that essentially started because people wanted to play around with these machine learning models back before people were really using machine learning in practice. So the AML folks are a pretty different set of people from the security folks, and mainly what happens in the AML community is that people publish papers. They drop them on Archive, and maybe they will show, they will put something on GitHub that shows *Hey, this is a particular attack* or, *This is a particular defense*, and there is this really quick iteration between attacks and defenses where once a defense gets put out, attacks very quickly break it. That kind of thing that is happening in the community. But there is essentially no discussion of coordinating these kinds of problems that are showing up in systems that people are actually using, except for a couple of different points. For example, one of the things that we referenced in the paper, when they released G52, they released it in stages, so that they would not be dropping this possibly dangerous machine learning model out there for people to play with and get hurt by. Then there is also some other efforts.

Jono, I know that you are working with some folks who are doing an AI-incident database, and how they can let people know about these kinds of things before people put these into products. But my understanding of those efforts is that those are more focused on, *These are the ways that these things can fail in general*, not *These are the specific ways that these things have failed. Let's fix these specific things in a specific problem.*

Jono: Well, that is right, and it mirrors the challenges between safety and security in general. There are a lot of, say, safety-focused people in the automobile industry. There are a lot of

security-focused people for the computers that operate in vehicles. Even those two groups of people don't talk terribly much. Because the heuristics that you need to use to find a security flaw are rather different than what you need to do to find a safety flaw.

Nathan: Even if a security flaw reveals a safety flaw.

Jono: Yes, so I think that Bruce Schneier talks about the difference between making sure the Tacoma Narrows Bridge doesn't oscillate and fall down in the wind, you know, the famous bridge. So basically, protecting the bridge against random variation from nature, versus protecting the bridge from a group of people who are trying to bore into it and plant charges to bring it down.

Nathan: Yes, those are pretty different.

Jono: You would do rather different things in your engineering of the bridge. If machine learning engineering also needs different approaches for safety and security that are different enough that they are different approaches, Allen, do you think that using some sort of common parlance to connect to the security community would help the adversarial machine learning folks to do that?

Allen: I think so, especially when we are talking about what the effects of a machine learning weakness might be. There are plenty of situations where you may have a machine learning system, it has a problem either in the data that it was fed to train it or in the inputs or people can mess with it, but maybe the impact of that is almost nothing. In cases where it actually does have potential safety impacts—self-driving vehicles or even semi-autonomous vehicles are really important safety impacts, so therefore, if the AI is able to cause a safety problem, then you do have a need to be able to communicate that with both the safety folks and with the security folks. A lot of this has to do with, traditional security vulnerability analysis would look at things like, *What is the attack surface of the software? What are the untrusted inputs that an attacker could potentially give it?* As Nathan mentioned, you could have stickers. Amazon sells a t-shirt that is just all stop signs. If I wear that, I can cause cars to stop on the road. Have I exploited the vulnerability? Well, maybe. There is a communication need there and, in the paper, we talk about this idea that came out of MITRE, which is actually part of the background for why CVE is a thing. With CVE, traditionally you have security analysts, and you have software developers. Those communities used to not talk to each other. Now you have things like DevSecOps where everybody is on the same team, and we are all trying to do the same thing. Back in the old days, that was not the case, so you needed a way to be able to say, *You have this problem, and we have a common identifier between them,* and they go into talking about boundary objects and trading zones. This goes all the way back to some sociology around, back in the old days when you were at a shipping port, and you had lots of people who showed up with different cargoes, and they all

speak different languages, but you needed to be able to exchange goods for money and so forth. So, you wind up with these boundary objects. Money is a boundary object where we agree on what money is. We don't have to agree on whether or not the bread is worth the same thing to each of us, but we agree on money. So, money becomes a boundary object. But even things like identifiers on cars and VINs and ISBN numbers and all of those IDs become trading things because an ISBN is useful for a publisher to keep track of what things they published. It is also useful for a bookstore to do inventory on which books are on the shelf and how many of each unit they have.

So, for vulnerability management where you are trying to track, *What bugs do I know about that are potentially security affecting? How do I know if I have fixed them all and if I fixed everything that I know about*? IDs assigned to that, especially common ones where, *If you are talking about this problem that you know of, and I am talking about this other problem that I know of, are we talking about the same thing or not?* That is where identifiers really come into use here. CVE is not the only identifier that is useful here in this space either, there is also CWEs, common weakness enumeration, which is more for like the classes of problems we might have. We go into that in the paper as well where we look at, most of it is talking about CVE IDs, but CWEs may also be an appropriate way of identifying categories of problems. I think since we put the paper out, CWE has even been updated a bit to start to address some of those categorization problems of ML.

Jono: I do not remember if I have it memorized, but it might be CWE1035 or 1039 [Editor's note: It is CWE 1039]. I do not remember. Anyway, there is one for machine learning stuff.

Nathan: Oh cool. I didn't know that.

Jono: Yes, it basically is very similarly defined to the all-things-trained-by-gradient-descent have a problem. But a little bit more general, which is interesting for our sort of CVE, CWE discussion if they have their CWE as something slightly more general than this one algorithm has a problem.

Nathan: What is weird about this one algorithm having a problem is that in traditional software there is a bug and it is often a line of code that we can fix to fix the bug. But in machine learning, we do not have a line of code. We write code to train these models. We do not have a line of code to change that fixes this problem. We do not know how to do it yet.

Jono: So, Nathan, is that an "our knowledge" problem or is that because you presume that there is a bunch of interdependencies, and they have to all be changed or it does not work?

Nathan: To reflect the question back to you, are you asking whether or not if we knew more about machine learning, like our machine learning science took off, we would know what to change? Or are you asking something else?

Jono: That is my question.

Nathan: I am going to give an opinion here. Generally, when I think of machine learning systems, I think of systems that are not specified. Like in traditional software, you go through requirements gathering, and you specify it, you write it down, you implement it, and if there is a bug, one class of bug could be that your implementation was wrong from this spot. That would be one kind of thing. In machine learning systems, they do not operate that way. What we do is we give them data, and then we train them from those examples in that data. So, even if the machine learning works perfectly in the sense that it exactly captured the patterns that we gave it in the data, it is not necessarily true that we constructed a dataset that would teach the machine-learning algorithm what we actually wanted it to be taught. So, there is this issue where training things and then checking what they know is just super hard.

Jono: Sure. But don't we get a little bit lost in the metaphors here? I do not train a machine learning algorithm like I train my dog. I write a piece of code that exemplifies the training methodology, and I can write some other code that exemplifies the data-gathering methodology, or expresses it, sorry. So, if the flaw were, *Use this algorithm instead of this algorithm*, I can change the import statement in my Jupyter notebook or whatever.

Nathan: One of the things that we talked about at the very beginning is, there are these different classes of attacks. We can make a machine learning system do the wrong thing. We can make it reveal the wrong thing, We can make it learn the wrong thing. Let us dial in on learning the wrong thing for a second. Let us assume that we had a perfect machine learning system that does not have any problem but what an attacker can do is they can modify the data in a particular way such that now when that perfect machine learning algorithm comes and learns, it learns what the attacker wanted it to learn, not what the person who was designing the system wants it to learn. So, does that help answer your question? Because that would not be fixed with, for example, a new import statement. The import statement here is fine.

Jono: But it would be changed by, say, changing the lines in the data that were injected by the attacker.

Nathan: The instances of the data, sure.

Jono: Yes. Which is still code to change, right?

Nathan: Well, it depends on how you are collecting the data, but sure.

Jono: I mean it is all code at some point, right?

Nathan: Uh…I guess images come off of a sensor, that kind of thing, sure. It is all code at some point.

Jono: Well yes, I guess there are interesting questions about the physical attack objects. So, Allen, would you, say, give the pair of glasses that might make me appear to a facial recognition algorithm as Tom Hanks rather than myself, would you say there was…

Nathan: You aren't Tom Hanks?

Jono: I know that I am very dashing, but I know that I am in fact not Tom Hanks. So do those glasses deserve some sort of identifier, like are those an attack? Are we going to have to start giving CVE IDs to physical objects?

Allen: Yes, so in the case of Tom Hanks' glasses or glasses that make everyone look like Tom Hanks, I think the actual question there is, what is the security impact of that use? So, if it is just simply that the Instagram filter that tags Tom Hanks in photos goes nuts, that is probably a problem for Instagram, it is probably not a security problem. It is a data-integrity problem for them, which they may treat as a security problem. But I certainly wouldn't think that I had been hacked if I had posted a photo of myself wearing these glasses and it suggests Tom Hanks as a label. However, if I was using facial recognition for identification and entry into a physical facility, and Tom Hanks was authorized to be in there, I am not, and I show up wearing my Tom Hanks glasses, and the camera takes my picture and the door unlocks and welcomes me in, yes, there absolutely is a security flaw in that system. But I think that would be localized to the system in which that flaw was deployed and not necessarily to the fact that that flaw is possible.

Jono: Great, thanks for those examples. You know Nathan, can you connect this a little bit to the other work you do around AI/ML and how that contributes to AI and ML engineering and what your broader goals for engaging with vulnerability coordination are?

Nathan: Sure. So one of the things that is currently happening in the field is that doing the machine learning and AI more broadly is not so much an engineering discipline as it is a bunch of heroic efforts. Someone will put together a really good team, that really good team will do some really good work, and you can list off where those teams are at major companies, and at major research universities and those kinds of things. But there is not a set of rules and practices where we can say, *OK, if you want good AI systems, do these several things*. Well, the SEI has put out a list of foundational practices for AI engineering, we have had an AI engineering effort that we are trying to kick off through ODNI [Office of the Director of National Intelligence], but we are still in the very early stages about how we make AI systems repeatedly and well. My major interest is about robustness of machine learning systems and their security. And I put those

in that order for a particular reason; right now if your ML system does not work very well, then it does not matter if an attacker is going to attack it because no one is going to deploy it anyway. To go back to the Tacoma Narrows Bridge example, you have to have a bridge that will not fall down just because the wind is blowing before you start worrying about protecting if from the people who are trying to blow it up. So, the vulnerability-coordination work is perhaps early, because in a lot of these domains, people just are not using ML for mission-critical stuff yet. But if we look at how things have been going in the last 10 years, we fully anticipate that both industry and the DoD will be using machine learning systems in mission-critical things, and we want to make sure that we have the infrastructure ready to respond to these kinds of failure modes.

Jono: Thanks. Allen, can you tell us a little bit about the vulnerability-coordination work with CISA [Cybersecurity and Infrastructure Security Agency] and how CERT/CC's role has evolved over the last—is it 32, 33 years now—of vulnerability coordination and the way that we have transitioned this sort of stuff to emerging-technology groups?

Allen: Sure. In three short sentences I can summarize 33 years of work.

Jono: I am sure that you can provide some links in the show notes for some of the more detailed things.

Allen: Yes, so the CERT Coordination Center started in 1988 after the Morris Worm, which was caused by somebody who wrote some code that exploited some vulnerabilities and very shortly thereafter, more vulnerabilities—I believe an FTP server—were discovered. That was one of the first things that CERT put out as an advisory back in the '80s. We can jump ahead 20-some years, and say that that continued to happen. During that 20 years or so up until the early 2000s, even large vendors really still were not taking vulnerabilities very seriously. They would kind of treat them as annoyances, but they were not necessarily really getting into patching and issuing updates and things just because of security vulnerabilities. That all started to change in the early 2000s. Folks like the Bill Gates security memo as being a real turning point for Microsoft, and a lot of the industry followed suit after that.

Around that time, DHS [Department of Homeland Security] was created. CERT/CC had been originally chartered by the Department of Defense to do this service for the whole internet. You know, *If there is a security problem on the internet, call CERT/CC*. There was a phone number you could call and get help. The internet got a little bit too big for that. Probably before we noticed it got too big for that. The roles shifted towards, rather than responding to incidents, it is more about figuring out how to teach a community of folks to respond to incidents and respond to vulnerabilities and building that community, the Forum of Incident Response Teams (FIRST) security teams.

Jono: Incident response in security teams.

Allen Incident response in security teams.

Jono: First.org everyone. Go check it out.

Allen: They have been around almost as long as CERT/CC has because it was a recognition even then that this was not the thing that you could have a handful of people in Pittsburgh taking care of for the whole planet. Our role has developed over time in conjunction with Homeland Security and the development of what initially was U.S. CERT and now has become CISA, as a separate agency within DHS. CISA's role is really to protect the critical infrastructure of the United States. The internet plays a role across all of the critical-infrastructure sectors. Go figure, people have computers to do their jobs no matter which sector you are in. All of those computers have software on them. Most software has vulnerabilities. There is a distinct connection there between, *Your ability to fix software vulnerabilities affects critical infrastructure*. So, CISA's interest is in that. They also have the responsibility for the civilian side of the federal government.

Jono: We saw a first wave of people having a hard time recognizing that vulnerabilities were a thing in the '90s for the general IT vendors. Then I think what you are saying is that we saw a second wave of that that is still going on with the operational technology people in the critical-infrastructure sectors following a similar growth pattern.

Allen: Right. All of the places where you used to have manufacturers who produced durable goods, and they added a little Wi-Fi chip to it, or they added a computer and system on a chip so you could have an app that goes with it, those become part of that attack surface for critical infrastructure as well as all of the operational technologies.

Jono: Yes, and this is like the smart light bulbs getting recruited to botnets, because they are running some ancient version of Linux because the people who make the light bulbs never set a way to update the operating system.

Allen: Right. So, suddenly every company in the world is now a software vendor as well as being whatever products they produce. They have to think about how they are going to deal with those vulnerabilities.

Jono: Do you see a similar pattern evolving in the way the machine learning community is growing and adapting to its own importance and distribution in the world?

Allen: I think ML is going to show up in a lot of places. Hopefully it does not come as a surprise to ML practitioners that they are using software, and that they might have software vulnerabilities, or vulnerabilities to deal with. But I think there is, as Nathan already mentioned,

there is a difference in communities here, where the communities are not necessarily used to speaking to each other on those terms.

Jono: Nathan, would you call the machine learning practitioners software engineers or statisticians?

Nathan: Are you trying to get me in trouble?

Jono: Yes. My point being, Allen made the assertion that the machine learning developers are software engineers, so clearly they know there are vulnerabilities in their software and how to manage it.

Nathan: So, you nerd sniped me, so you have to content yourself with my response. Machine learning and statistics are two different fields that have their own lore and their own way of talking and their own way of doing things. I am myself a statistician, my PhD is in statistics from Carnegie Mellon, so I am sort of from this side. When I introduced myself, I introduced myself as a machine learning research scientist. Modern statistics and modern machine learning basically do the same stuff but we think about it very differently. So, the machine learning people who do machine learning research don't think of themselves as software developers, they often think of themselves as computer scientists in the sense that a computer scientist is different from someone who writes software. Similarly, for statisticians, statisticians often think of themselves as applied folks who will write the code that they need to write to get a job done. But neither of those communities really think of themselves as developers. That is why we have fields like, other titles, like data scientist on the statistician side or ML engineer on the ML side.

Allen: I think there is an operationalization that is going to occur across ML because right now there is a lot of theoretical work, and there is some actual operational work that is starting to happen where companies are deploying ML models into production systems today. The application of that will only continue to increase, so I think as it gets operationalized, it is going to have to go through a similar transformation to what IT operations has done, where we went from in the '90s and earlier, it was possible for individuals to log in to individual machines and type commands and upgrade software and things. Now you have scripted and orchestration, and you have all sorts of tools to help you do that and lots of engineering thinking has gone into, *How do you build robust systems out of non-robust components in traditional IT?* A lot of that came from the fact that having fingers on keyboards does not scale to thousands of computers within an organization or hundreds of data centers. That is the scale at which computing and ML stuff is going to be deployed. Some of it also is because ML models will change as the data changes and as they get retrained and everything, so they are already going to have to think about this idea of, *If there is a problem with my model, and it is costing me money, I need to be able to fix it and redeploy a new thing very quickly*. Well, that is the same pipeline you want to have for

security problems anyway. The mechanical parts are going to show up just because of the fact that ML in its importance will become critical, and therefore having it be maintainable will become critical.

Nathan: That is not to say that there are not ML product owners and ML engineers and ML research scientists that all do these different tasks. I didn't mean to imply that.

Jono: So, we have this thought experiment or two really of, what if we apply CVE IDs to these different parts of the ML pipeline. What would each of you like a reader or one of our audience members to take away from that work? What is the thing that people should know and take away and change what they are doing?

Nathan: Major takeaway that I would offer to people is that if you are using machine learning, you are using it to solve a task, and the reason that you are using it to solve a task is because you want something done. If the doing or not properly doing of that task has an implication for your security posture, then you need to think about what the failure modes of the machine learning system imply for security. An ML system is not secure. It is secure for what purpose? You have to think about your security policies, and then you have to think about making sure that those security policies are met when you introduce this new shiny ML thing into it. That is the main takeaway that I would want someone to have from the paper.

Jono: Great, thanks. Allen?

Allen: I am just going to echo what Nathan said. I think understanding your threat model, understanding what you need the system to do, what its attack surface is, and having a good understanding of what the implications are if someone messes with that, is really the most important piece here. We have talked about how there are lots of applications where ML may be deployed and doesn't have any security implications at all. You may have quality problems, but it is not security related. And one last thing, because these things are getting deployed in so many places where safety is an issue, the SEI has done some work on the intersection of security and safety as well. My personal opinion is that you can't be safe unless you are also secure, because if the security gets messed with then your safety claims go out the window. So security, safety, and robustness are really very tightly coupled, and the system can be engineered to be all three.

Jono: Thanks. I know that we have produced a guide for machine learning and cybersecurity. We have this other work on vulnerability categorization and prioritizing vulnerabilities with SSVC [Stakeholder Specific Vulnerability Categorization]. Nathan, you mentioned your other work with robustness and security stuff for AML and the AI engineering and the various other things. What is next for our work in this area in the intersection between ML and vulnerabilities, do you think?

Nathan: I think getting people to join us is a lot of what is next. For example, Microsoft and MITRE are certainly thinking about these things. The folks that you have been talking with, Jono, about the AI-incident database, are thinking about this stuff. There is a lot of community building I think that needs to happen so that we can achieve some critical mass and actually get some things done. It is not that we do not know what to do, it is just that we have not done it yet.

Jono: Allen, what do you think?

Allen: I think there is a good bit of bridging to be done still where we have communities who are using different vocabularies to talk about similar things, and in some cases they are using the same vocabulary to talk about different things, and that is actually probably the more insidious problem to be aware of is the unrecognized mismatch between, you use a word and I use the word and we both think we are talking about the thing that we think we are talking about, but it turns out that your idea and my idea are different. So, building some of those common vocabularies, common concepts, and getting people on both sides, the ML and security side, to understand those things. Again, coming back to this idea of creating trading zones, I think that is really where community building will help do that because it brings more people into the room to start to develop those vocabularies and that understanding.

Jono: Yes, and just one—Nathan, you mentioned this—but Rahm's work with Microsoft and that community building there, we will put a link to that stuff in the show notes as well. Allen, Nathan, thank you so much for joining us today and talking to us about a very nuanced topic, I think. For all of our listeners, there will be links to all the resources that we have talked about today in the show notes for anyone who wants to dive in a little bit more detail. Allen, Nathan, thank you very much.

Thanks for joining us. This episode is available where you download podcasts, including SoundCloud, Stitcher, TuneIn Radio, Google Podcasts, 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 do not hesitate to email us at info@sei.cmu.edu. Thank you.