A Stakeholder-Specific Vulnerability Categorization

Featuring Allen Householder and Eric Hatleback as interviewed by Jonathan Spring

---

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: My name is Dr. Jonathan Spring. I am a senior vulnerability researcher here at the Software Engineering Institute's CERT division, CERT Coordination Center. I am joined today by my colleagues Allen Householder and Eric Hatleback. Today we are going to talk about an approach to prioritizing vulnerability response that we developed here at the SEI that we are calling stakeholder-specific vulnerability categorization [SSVC]. Welcome.

Allen Householder: Thanks.

Eric Hatleback: Thanks.

Jonathan: Let's start off by having you tell us a little bit about what you do here at the SEI day-to-day, that sort of a thing. Allen, I think you have been here the longest, so we are going to give you the honor of going first.

Allen: I am the technical lead for threat ecosystem analysis, which is a long way of saying that I do a lot of modeling and mathematics, related tasks associated with either vulnerability categorization, sometimes it gets into malware clustering, and other things like that. Also looking at coordinating vulnerability disclosure and some other large-scale, macro-scale problems within cybersecurity and vulnerability disclosure and vulnerability analysis.

Jonathan: Great. Eric, go ahead.

Eric: My background is actually philosophy of science. I have a PhD in that, so some of the technical stuff is left to the colleagues often. But I am on Allen's same team, the Threat Ecosystem Analysis team.

Right now, though, what I am spending most of my time on is vulnerability coordination: straight coordinating with the team, taking cases, coordinating with vendors and researchers, and trying to get vulnerabilities patched before they get out and wreak havoc. So, that is kind of what I am up to at the moment.

Jonathan: Well, yes, reducing havoc seems great. I also am on the same team. I have been working on the vulnerability-categorization stuff as well as some machine-learning aspects of things related to incident response, incident-response automation, and philosophy of science. I did some network forensics and things like that in the past, but not so much anymore. Although I was at FloCon talking about those sorts of things somewhat recently.

Let's talk now about vulnerability management at a little bit of a higher level to introduce that. What is vulnerability management, first briefly, and then how or what are the current tools for it, like namely, how does the Common Vulnerability Scoring System (CVSS) play into vulnerability management?

Eric: Want to go first, Allen?

Allen: I think vulnerability management actually means a few different things to a few different people. In our situation, we really talk about it as the thing that happens at the vendor, who produces the software and needs to fix something. They need to analyze reports that they receive and triage them and do things, make decisions about how they are going to prioritize their efforts. Then they release patches. There are folks who have that software deployed in their networks, and they need to deploy that. They also probably need to make prioritization decisions about what they are doing, what they should do next, and how quickly they should patch things because it is not always possible to patch everything. There are also some aspects of scanning for vulnerabilities and doing penetration tests and whatever to find out what is wrong in your network and going and remediating those things as well. That also involves prioritization decisions on those reports. I think those are the three main aspects.

Eric: Yes, I think there is the other side too, like the coordination side that I am doing now and the researchers as well. To get the vendors the information they need that there is a vulnerability, often that takes somebody, a researcher finding that in the first place. Then what do they do with that? Do they send it straight to the vendor? Do they come to someone like us to help them coordinate with the vendor to get the vendor the information, et cetera, et cetera? The other side apart from just the vendors and their fixing is the finding and, *How does it get handled before other people know about it?* sort of things.

Jonathan: The different people or roles involved in this vulnerability management sounds really important: the vendor, the people who run the software that they buy from the vendor, the researchers, the coordinators. How is CVSS involved in that, or what is CVSS in the first place?

Eric: Broad level?

Jonathan: Yes.

Eric: I guess CVSS is the industry standard at the moment anyway for ranking the severity of vulnerabilities, I suppose. It is a single number that is spit out of an equation, and it is meant to tell people how severe something is supposed to be. I think that is my high-level take on it.

Allen: Yes, and shifting into what one of the issues that we saw that started some of this work is it is *a number*. There are eight or ten different variables that go into producing that number. It goes through some complicated math and comes out the other end as a number, but then it is the same number for everybody. Vendors might have different decisions than the network owners or any of the system owners, or the coordinators, or the researchers, too. Everybody has their own priorities, but they are all using this same number. If you just sort everything by a CVSS score and stack them, then you might wind up choosing something that should not be your highest priority.

Eric: According to your priority, right.

Jonathan: I know that we talked about this in a prior podcast that people can go look at and listen to for background on what is a problem with that. But is the equation that you talked about, is it clear how that was developed?

Allen: There is certainly a description of the process, which involved the first CVSS special-interest group in the FIRST community (Forum of Incident Response and Security...)?

Jonathan: Incident Response and Security Teams, yes.

Allen: The Forum of Incident Response and Security Teams. So, they took a set of vulnerabilities. I believe they decided that these are the vulnerabilities that they wanted to evaluate using input vectors. They decided what priority those things should have, what severity those things should have, rather, and then fit a model to that. The way that that model was fit is not entirely transparent even in the documentation they have on what the formula…

Eric: Taking the inputs, the 10 or so inputs that you mentioned as identifiable things and transferring them into numbers, somehow stuff gets lost in the shuffle.

Jonathan: Allen, you made a little bit of a quibble there between severity and priority. I think maybe the difference between those two words maybe is the lead in to what we have decided to do.

Can you introduce the stakeholder-specific vulnerability categorization [SVCC] and what we are proposing to do to support vulnerability management, maybe in addition to CVSS or instead of, depending on...

Allen: In terms of splitting between prioritizing and severity?

Jonathan: Yes, so in vulnerability management what is the difference between your priority about how you are going to act on the vulnerability and the severity of the vulnerability, and how does that inform the high-level bit of our work.

Eric: So priority I think has a lot to do with actions, which I think as we mentioned in here is what we take as our take-off point, which is you have a priority of things, that is the stuff you are going to actually do. Severity is somehow a measure of the thing that you are talking about, so I think there is a big split right there. We are choosing to focus in this work on the prioritization side of things rather than the severity. Severity has a part, but it is not the focus. It is not the entirety of it, as it is with CVSS.

Allen: I think too severity has a certain connotation, especially if you go back a decade or two when CVSS or the notions that evolved into CVSS were really being discussed. About the worst thing you could do at the time was take over a machine. So, most of that is geared towards how close is an attacker to being able to completely take over a machine, which is a great model for circa 1999.

Jonathan: Machine, do you mean laptop or desktop-computing device, not…

Allen: Right.

Jonathan: ...an auto-plant assembly machine or airplane?

Allen: Or an airplane, yes. Or an election system. Yes, so there are lots of macro-scale impacts that can happen that do not necessarily involve taking over a machine. In fact, data breaches were not really much of a thing when all that occurred. One of the things we found with internet of things [IoT], especially when you get into medical devices and more safety-critical systems including industrial-control system things, is that availability is actually probably more important than confidentiality. Really, you want the machine to continue working even if it is leaking data all over the place, because the machine is doing some critical function to keep somebody alive or keep somebody safe. So those sorts of impacts are not terribly well considered. They are not as clear cut in CVSS.

Jonathan: So, how does our work address the transparency issues that we just talked about with CVSS?

Eric: To start going into it, we are choosing to use decision trees as the method by which we will prioritize things. The transparency is pretty clear because for each of the decisions we are making on these trees, it is very clear what we are meaning by which choice we are making along the tree. I think that is the first step to the transparency answer is, *Here are our branches. Here are our options to choose from when moving our way down a tree*. At each point, we have some pretty clear rules for describing exactly why you would choose branch one, branch two, etc.

Allen: Those trees were developed by humans sitting around talking about what the decisions were and what the granularity of the decision actually is. They are not decision trees that are just fit on a big data set, and you run some decision-tree algorithm over it and get a decision tree out of it. It is actually based on experts sitting around a table and understanding the problem.

Jonathan: Can you maybe walk us through a simple example? I think that probably trying to quote a vulnerability off the top of our heads might be a little bit difficult, but there have been some important ones over the last bit of time. One that comes to mind maybe is the CryptoAPI DLL [Dynamically Linked Library], the Windows failure to authenticate crypto certificates properly, right? If I am trying to decide whether to update my Windows machine, maybe Windows server that is running, say, the Kerberos active directory stuff. I am the patch applier. What decisions, what questions do I have to answer in SSVC in order to make a decision about how much effort I need to put into patching that system?

Allen: We should also say that we actually do have different decision trees for both patch appliers and patch deployers.

Eric: Right, which gives us the *stakeholder-specific*...

Allen: Which gives the stakeholder-specific...

Jonathan: So, Microsoft has already issued the patch.

Eric: Right, it is out there.

Jonathan: They are already the patch deployer. They have decided to do it. It is Tuesday.

Eric: On their own, they would have done their own thing on their tree if we weren't at this point in the example. Now we are at a different point, which we are pretending to be applier, *Should we apply this patch ahead of others or not?* Things like this, yes.

Allen: The first question that our tree asks is about exploitation. That really comes down to, *Are there reports of adversaries actively exploiting this? Are there known exploits available even if they are not being actively used?* Things like, *Does it show up in Metasploit? Does it show up in Exploit DB or anywhere publicly?* Or, the other option is, *None*. *You don't have any knowledge of it*. Depending on those, which of those three options you are on, you then move on to...

Eric: It is *exposure* in each case, yes, after that.

Jonathan: I think that there is a proof of concept for how to structure a certificate that will be verified incorrectly for that CryptoAPI DLL. So, there is a proof of concept at least, and let's go with that for now. This may change, of course, by the time you are listening to this, everyone, but that is part of the point.

Eric: Yes, this is fluid, right? We can adjust as needed, get on a different tree if you have to.

Jonathan: Well, actually exploitation should be the only one of these four questions that changes over time.

Eric: I suppose that is right, yes.

Jonathan: But, anyway, let's say that it is proof of concept for now. Then, where do we go?

Eric: *Exposure* is next, at least as we have it listed. The good thing about the trees, as we talked about when we made them, was there is not really an order to these things, right? They could be placed in any order whatsoever. In the paper itself, we have *exposure* next, but it could have gone to *mission impact* or *safety impact* next. But this is the path that was chosen for the trees as we outlined them in the paper itself. The next thing you will evaluate is exposure. I guess I would say the exposure is how well…

Jonathan: So, *My active directory server, is it connected directly to the internet without a firewall in between it and the internet?*

Eric: Or not, so the level at which it is accessible would be a way to describe it.

Jonathan: Exposed?

Eric: Yes, exposed, right, to describe the exposure. You have *small, controlled,* or *unavoidable* as your choices, in any particular case. Then you will move to the next.

Allen: In this case, if it is an active directory server, that is probably controlled. It is not like a web server or a DNS server, which would be uncontrollable...

Jonathan: Right, its job is to connect to the internet.

Allen: ...its job is to be talking to the internet all the time. Active directory has to be exposed to your internal network, but it also does not have to be exposed to the internet. So, that is more controlled. And also, we have *small*, which I think is more about the either air gapped or even just tight, very tightly controlled enclaves within the network.

Jonathan: But your active directory server is probably only one hop away from the internet, and that hop is a cluster of humans clicking on emails connected to the internet [who] all have to go talk to the AD server. So, that cannot really be said to be small.

Allen: I would go with *controlled* for this example.

Jonathan: Yes, so I didn't tell you guys that we were going to score one of these on the fly.

Eric: See how easy it is? Right, so if you're moving your way down. We have a proof of concept in place as far as exploitation goes. We have got *controlled* as far as the measure on exposure. The next thing on the list out of the four things is *mission impact*. This is where it will really be specific to the company or vendor that you are, right?

Jonathan: Organization.

Eric: The organization that you are. That is right, yes. Abstractly it is actually hard to answer this because we do not know who we are in this case. There are a selection of options here describing how severely or minorly I guess your particular organization is going to be impacted by this.

Jonathan: Right, so we need to know what the organization's mission is before we can say what the impact on the mission is, of course. But the AD server is what lets everyone log in to all their computers. So, if your organization needs people to work on computers, it is going to probably be at least supporting your mission-essential functions to have the AD server running. But, also, unless it is your organization's job to just run computers for people to log in to, it is probably not actually your mission to run them.

Eric: I guess just to run down the options so that we can make some clarity in terms of how you are delineating what is going on here, our choices for mission impact are *non-essential degraded*, *mission-essential functions crippled*, *mission-essential functions failing*, or just *mission fail* altogether. Those are the strata by which we will try to choose how this would affect any particular organization. Or, if you are an organization doing this, your organization, of course.

Jonathan: Allen, I know that I sort of stole *mission-essential functions* from FEMA [See Annex B of this paper], but can you talk a little bit for the people who are not in the federal civilian government space, a little bit about what that means and how we translate it to regular private-sector industries?

Allen: It works its way up from low impact to high impact. So, mission failure at the high end is, the organization ceases to exist or is unable to achieve its mission. For example, if a hospital had to shut down even temporarily because of the impact of a vulnerability, then that would be a mission failure at least for some area of time.

Mission-essential function failure would be some function that is provided to the organization by the system, but not necessarily enough to make it completely collapse. But, you definitely are in a situation where if it does not get fixed soon, you are going to be at risk of more cascading failures. *Mission-essential functions support crippled* is somewhat less than that. I do not quite recall what the distinction was between that one and failure.

Eric: It is that the supporting functions are stopped, but that it does not actually stop the mission-essential function from happening right now. But that situation probably cannot continue. You are on backup power basically. The generator is going to run out of fuel eventually.

Allen: Yes, so possibly loss of redundancy. You are operating in a sort of degraded situation. *Nonessential degraded* is really one where some function that is not necessarily mission critical is down or degraded or temporarily disabled, or whatever. You can limp along with that condition indefinitely and not have to worry about it for too long.

Jonathan: So, if we are an organization that has not figured out what our mission is, FEMA, and we cite this, has guidance on how you elicit mission-essential functions from your mission and all of this stuff, which people can follow through on [For more on eliciting mission-essential functions, see appendices B, C, D in this FEMA paper]. But they shouldn't have to do this for every vulnerability, right? Once we do this once, we should know what the mission impact of our AD server is for every vulnerability that might be in it. Is that right?

Allen: Yes, I believe it is either at the individual server level or even at the service level across your enterprise. You make this decision once for that service or server, and at that point you know the answer until that changes. That will change far less frequently than you will be dealing with vulnerabilities.

Jonathan: From what you said, I think that this takes us through the middle path, again through *crippled*.

Eric: *Crippled*. Right, sure. Yes, we can go that route. Then, finally, before we actually reach our answer, we will assess the safety impact of the vulnerability. Our choices are numerous here. In fact, lots of these on our tree, for space reasons, are collapsed into all others. At the higher end you have *catastrophic* and then *hazardous*, and they tick down from there. How many we [have], I will flip back...

Allen: Five.

Eric: Five of them. So *catastrophic*, *hazardous*, *major*, *minor*, and then *none*. So, each of those in the paper itself is laid out with the pieces that describe that one accurately.

Jonathan: We adapted these from both the Federal Aviation Administration, if I can remember how FAA expands, but also with some insights into well-being as defined by the Centers for Disease Control. So, we have used as much as we can other peoples' assessments of these things. But, as bad as the thing that lets everyone log in into their email is getting compromised, it is not the sort of thing that causes airplanes to fall out of the sky. Although this is probably bad for the organization, the mission, it might not be so bad for human well-being.

Eric: Right, for the safety. Yes, the safety.

Jonathan: Is that difference there?

Allen: That and also we should note that we took a fairly broad definition of safety in this report. So, we might be using it differently than folks in the industrial control systems world would use it, where safety is really well defined in that environment. In our case, we include that definition of safety in ours, and I think these categories also track with those categories that you would run across there. But, this also includes ideas like financial safety or even psychological risk or bodily harm, lots of things up to and including, at the higher end, it is more like societal impact.

Eric: Yes, rather than impacts to individuals, it is impacts on systems of individuals and groups.

Allen: Or even entire services provided to a population of people or whatever.

Jonathan: That is how someone who is deciding to apply a patch would do that. That is where we've got the document now. I think that one of the things we are looking for is feedback on those details so that they are right. But, how hard would this be to try to automate?

Eric: Well, I think first let's finish…we should follow up and finish what we had. So, following through each of the nodes, what we would go down to, our *exploitation*, *exposure*, *mission impact*, and *safety impact*. The *safety impact*, as you noted, falls into the lower category here. That is going to fall into these *all others* on the particular chart we've got here at hand. That means that it is going to end up being a scheduled patch for the organization, the mythical one that we have in question here. The other options in case there were different vulnerabilities or different organizations. They run from *defer*, which is just, *Do nothing*. *Scheduled*, which is, *Do it when you would normally do these things. Don't break stride to do this. Out of band*, which means, *Yeah, break some stride and kind of do it as soon as you can*. Then *immediate*, which is, *Drop everything and get this done*. In using a tree, that would follow through with the example I had and tell you exactly what you are going to end up with. So, all of our final results there are one of those four, which hopefully helps either the patch developer, which we were not going through, or the applier, which we were. Tell them when or how soon they need to actually address the issue at hand.

Jonathan: Thanks for going over the suggested meanings of those. Of course, we are not trying to bind anyone to those meanings, but those are our suggestions. That is a good example of how it works for the applier. I do not know that we need to bore everyone with another one for the patch developer. I think that everyone sees how this goes.

What else do we ask people to bring to the table maybe that is slightly different from CVSS? Is this more or less effort? Or, what is the benefit to the slightly increased effort to the patch applier? What is the down-the-road benefit if we do get this a little bit more refined and automated? What are the potentials for automating some of this and integrating with asset management and these sorts of things?

Eric: Well, to the first part, I mean as far as difficulty or not, I guess it probably would be a little more difficult than CVSS, because the CVSS you essentially plug in the stuff on the webpage, and it will give you your number right there. It is a bit of a *just plug in and go*. Here you need to think about your specific organization, but that is where you get the benefit of the little bit of extra effort, I think, because it is very much tailored to you. At least that is my view on it.

Allen: Yes, and most of the hard work comes in the mission impact and the safety impact, which really again, those are system-level things. They are not per vulnerability. So the things that change over time with each vulnerability is really the exploitation and, I guess, exposure somewhat, but even exposure is more particular to the service. So, you might be able to go through in one assessment and figure out what your exposure, your mission impact, and your safety impacts are for all of your systems. Once you have that, you have it. Now that just becomes a lookup table. Then what you need to know is, *Well, is this thing being exploited today or not?* or *Are there POCs available?* That you can get fairly readily from as simple as, just pay attention to Metasploit and Exploit DB and use that as a gauge. Or, pay attention to Twitter and maybe subscribe or pay for a threat-intelligence feed or something like that. That probably becomes available to you as a data feed one way or another as well. Now you can just take your inventory and your threat-intelligence feed, and if those are in a decent format where you can extract this data easily from them, now you can just combine them up and get the values for those four variables and look it up in a table, and you have your answer.

Jonathan: The question that I have been getting a lot, as I have talked about this a few different places in January, is this ready to be used? As a federally funded research and development center [FFRDC], an important role that we serve is an honest broker of information, a way to do things that is not biased by a profit motive or something else. So, what do we need to do to get this ready for anyone to use?

Eric: More feedback is probably the main thing, right? As we are pretty clear about here, this is the proposal, right? We are proposing this, and we have run through a lot of hours of discussing and internal[ly] testing the ideas and running through it. But that is not going to be a substitute for a wide-scale, *Hey, everybody, look at this thing and let us know what you think*. I think first and foremost is more exposure and more feedback about what we [have] here.

Allen: Our evaluation has largely been through role playing. We have…

Eric: As we did here, in the example earlier.

Allen: Right. We have each brought some experience in the outside world to the table with different organizations and how they do vulnerability management. We have walked through scenario-based role play to evaluate a number of vulnerabilities. That is not the same as people who actually have their fingers on keyboards and can deploy patches in the real networks and have to pay the price when that patch goes badly. We really do need that feedback from operational folks to give us that information.

Eric: And, the actual context of actual vendors and what they would do, because we are saying, *Imagine we are this or such*. They will know exactly what they have, what they are running, how they need to deal with it, etc. So...

Jonathan: So, is what we have published with SSVC ready for interested organizations to beta test and give us feedback on?

Allen: If they have the data that they need to input to this or they can easily get it, then, yes. I think. If you [have] a network inventory or a system inventory that you can apply to this and you have threat-intelligence feeds, you can potentially manually do this. If you want to try some automation, we have put a little bit of Python scripting out on GitHub, which we will have a link available for you, that does the lookup for you, assuming that you [have] the data to input to it. So, it does not do anything to help you figure out what your answers to those four questions are, but it does…if you have those answers, it can give you the lookup table and give you an answer. So that is available for folks who are ready to be able to integrate that.

Even if it was just a tabletop exercise that someone decided to do at an organization to see, *Does this look like it would work for us*, we can still use that feedback, too. Just because, like I said, the more experienced folks we have who look at this and try to use it or at least evaluate it, the better off we are going to be able to improve it over time.

Jonathan: Think we will put a contact link in the show notes for anyone. Sounds really good. So, given all of that and the work that we have put into this over the last year, year and a half with talking about what CVSS needs to have changed and then what SSVC can bring to that change and sort of move the process along, what are the future areas, like next future areas of work on prioritizing vulnerability management?

Eric: Well, one that I am actually interested in, given what I am doing with the coordination stuff right now, is developing a tree for coordinators. It was on our table initially. We had it in here, but it blossomed on us, all this, and we decided, let us put that one on the back burner for next time. I am really interested in getting to that. First and foremost, at least from my perspective, is working on the tree that coordinators might use to work with this sort of stuff.

Allen: I think too we have talked here about the system-owner version of the tree. We have not really talked so much about the patch-developer version of the tree, which do have some different variables in them, which we also need feedback on as well. One of those is technical impact, and the other one is attacker utility. So, *What does the attacker gain by exploiting this?* We break that down into really two dimensions of, *Does it make attacks very efficient,* or, *Do they gain a lot of value by exploiting a few machines or a few systems,* or is that more diffuse in that they might…. So concentrated value is, *I can break into the database and steal the entire database*. Diffuse value might be, *If I hit 1,000 machines, I can build a botnet out of 1,000 machines and now have complete network capability*. So that is what the attacker utility means.

Understanding how that works on the developer side as well is going to be important to us, too. I think there is definitely an opportunity there, not just for feedback from system owners but also from the folks who develop software and have to prioritize their incoming cues. CVSS itself actually might fit into the technical-impact portion of this whole thing. So, understanding how does this fit in a context where CVSS continues to exist and people are already using it, but maybe you also want to add this decision-tree mechanism on top of that. Understanding that is another angle that I think we can go forward with this.

Jonathan: Great. Well, do you have anything else that inquiring minds need to know about vulnerability management?

Allen: One point we should make is, if you are in the very fortunate position to have enough capacity that you can patch everything, that is what you should do. You do not need a prioritization scheme. You do not need severity scores. If you can service all of the vulnerabilities that come in your front door and fix them, do that. It is only when you do not—and unfortunately, I think almost everybody is in that second category—only when you do not that you actually need to be able to prioritize things. Don't forget the fundamentals. If you have the capacity to do it, just fix stuff.

Eric: I do not have anything specific to add at this point in time. I think that is right. This is a prioritization scheme, right? So, if you can do them all, you do not need to prioritize anything. You can just do them all.

Jonathan: But it is important that the recommendation is probably that if someone decided that it was important enough to fix the software, it is probably important enough for you to apply that fix.

Eric: Yes, do not defer just because there are patches out there. I think Allen's describing a situation where you have the capability, the resources, the time, all that stuff to actually apply all the patches. Do that, for sure.

Jonathan: One of the things that I have heard a couple of people say is that a transparent decision-making process about which things are important, if you get management to buy in, they are like, *Yes, this is our risk appetite and this is the service level of quickness that I would like to be able to patch something that is this bad*, right?

If your vulnerability-management team or your SOC or whatever it is, cannot keep up with the pacing that comes out of a very transparent decision-based process on risk, you have very good evidence to go to management and say, *We need to hire more people*. *You have said that this is your risk appetite*. *You have said that this decision process matches your risk assessment, and we are not keeping with it, so we need more resources*. There are both sides of this being potentially useful. If you do not have enough resources, you need a way to tell people.

Allen: I think that goes back to the, *How is this more transparent?* One of the transparency pieces there is it is not only explainable, but it is also understandable by senior executives who have a sense of, *Well, if this happens to my business, that is bad. I understand how bad that this, and I therefore want to do that*. *This affects my safety profile, and I understand how bad that is*. So it helps you to translate some of that into something that both the folks on the front lines handling vulnerability management stuff and the folks in the board room can understand, *This is why we are making these decisions about these vulnerabilities*. If you are wrong, you can have a meaningful conversation about, *Well, okay, which of these pieces are we not connecting on?*

Eric: Versus just what does that number mean. Like, so I don't know what that...

Jonathan: Yes, versus why did you not patch all the 7.3s?

Allen: Which doesn't necessarily have much meaning to someone who is not familiar with CVSS.

Jonathan: No and of course in the way that CVSS is done, it is not always obvious that what they are doing is sort of a fastest plus fastest equals two. So of course, it is not understandable, because that is what they are doing.

Eric: Yes, turning the scripters into actual things, into numbers almost arbitrarily leaves you with an arbitrary number that you might think means something different than I think it means.

Jonathan: Great. Well, thank you for being here and talking about this work. To our listeners, thank you for joining us today. We will include links in the transcript to all of the stuff that we have talked about, all the various resources. We also include, of course, a link to the white paper with SSVC defined in all of the gory details. There is also a blog post that Allen put out and you will be able to find a link to, that is sort of a short summary to, what we have talked about today.

> [Editor's Note: *Since the recording of this podcast, SSVC has also been accepted for publication at the Workshop on the Economics of Information Security (WEIS) 2020. A version 1.1 of SSVC, with minor updates, has been published there:*
>
> *http://weis2020.econinfosec.org/wp-content/uploads/sites/8/2020/06/weis20-final6.pdf*]

This podcast is available at the SEI website. If you are listening to this, you already know that, that is where you got it. As always, if you have any questions, please do not hesitate to email us; info@sei.cmu.edu will get to us. Thank you.

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 don't hesitate to email us at info@sei.cmu.edu. Thank you.