CERT Guide to Coordinated Vulnerability Disclosure

Featuring Allen Householder as Interviewed by David Warren

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.

David Warren: Hello. My name is David Warren. I am a senior vulnerability researcher here at the SEI. I'm sitting down today with Allen Householder, who is also a senior vulnerability researcher at the SEI, and we are going to discuss The CERT Guide to Coordinated Vulnerability Disclosure..

Welcome, Allen.

Allen Householder: Thanks.

David: Let's talk about a little bit of what it means to be a vulnerability researcher here at the SEI. Can you describe how you came to work at CERT and what you do?

Allen: My background way back in the '90s was a network and web-infrastructure engineer. I came into the CERT/CC as an incident analyst back around the time that Code Red and all the worms were hitting and started getting involved in some of the analysis of how worms spread and how malware spreads and things which led to doing some work with the malware analysts. We worked on packer signatures and then malware trends across the malware database.

Then, eventually, I wound up doing vulnerability analysis, in particular working on the fuzzers that CERT put out a few years ago. In the process of doing fuzzing, we discovered that vulnerability disclosure was a big bottleneck because we could find lots of vulnerabilities, but we couldn't necessarily coordinate and get them fixed. Fuzzers are really good at finding stuff. They are not so good at making people fix things. That sparked my interest in coordinated disclosure. Since then I have started to take a broader look at the ecosystem in which disclosure happens and all of those processes that go along with it. So that brings us to today.

David: I remember that. I worked on the fuzzing aspects with Allen. I remember generating the thousands and thousands of crashing test cases that we would then send to a vendor and then they would have to ingest and deal with. Sometimes they weren't as thankful as you would think for people that were doing QA [quality assurance] for them. But also, it would lead to finding vulnerabilities in libraries that are used in lots of different software that you would need to coordinate with lots of different vendors, which led to coordinated vulnerability-disclosure situations that we had to document and make policy. Exactly through that experience is what motivated the CVD guide.

Allen: Very few software vendors are really happy when you say, *Here are 300 more things to put on your to-do list*.

David: Sure, absolutely. Each one of those became a ticket in their bug-tracking system that a software engineer had to look at. I sympathize with them that it was expensive. Because they are security issues, they were sort of forced to look at them. We documented our process for other folks to follow and to lay the groundwork for doing coordinated vulnerability disclosure.

Let's talk about what coordinated vulnerability disclosure is. What is it, and why is it important?

Allen: Basically, it all starts with somebody finds a vulnerability in a product or service. That might be because you have run a fuzzer, and you have found some result and determined that that is likely to be exploitable. That could be because…there have been vulnerabilities reported in Xbox where a little kid was just mashing the buttons on the controller, and it turns out he was able to buy something through a store. There was vulnerability or something in the service. People can find vulnerabilities in lots of different ways.

Once you have found a vulnerability, now the world divides into two groups. There are the people who know about the vulnerability and everyone else who does not. So, the question is, for people who know about the vulnerability, you have to start from the standpoint of, *OK, what do I need to do about this?* and, *Who else needs to know?* and, *What do they need to know about it?* and, *When should we tell them?* You just iterate over those questions until the answers come back as, *There is nothing else to do,* and, *We have told everybody who needs to know to be able to do something to fix it*.

Usually that means researchers will tell the vendor or the service provider about the vulnerability. They will work with them to deal with, *Here is what the problem is. Here is what we think the fix is*. The vendor or whoever the responsible party is that can modify the software or the system to fix it, does so. Then the researcher may check to confirm that that has been fixed. Then there is a process of, sometimes that also results in the need for public disclosure where there is something further that needs to be done. If a vendor of a shrink-wrapped product

[off-the-shelf software][1] fixes something, it might also require the users to go patch and apply patches or something. It does not always result in public disclosure because sometimes it is a service. If you report something to a service provider, and they can just fix it and deploy that fix across all their systems, there is really no user interaction there. So, you may or may not need to provoke the end users to do something. In that case maybe public disclosure isn't the right thing. But the coordination piece and then the disclosure piece…disclosure happens in stages to different people and different parties at a time. The disclosure piece does not necessarily mean publication, but sometimes it does. It definitely means telling the people who are most likely to be able to fix the problem and getting them to the point of actually fixing the problem.

David: So, you are optimizing for societal good if possible?

Allen: Yes. You are trying to get problems fixed before they become actual problems. Obviously, vulnerabilities are situations where systems are exploitable by adversaries. You would like to be able to get those fixes deployed to all of the affected systems as quickly as you can, so that there is less opportunity for adversaries to leverage those and do whatever it is that they do.

David: So, minimizing the window for attacks.

Allen: Yes.

David: Great. So to that end, CERT has released the actual guide to coordinated disclosure. Would you like to go into the background of the original guide and then how it has been used since the initial publication?

Allen: CERT has been doing vulnerability coordination since just about 1988, when it was formed. One of the very first CERT…in fact, I think *the* first CERT advisory was an FTP-vulnerability advisory in December of 1988.

David: Stack buffer overflow, I would guess.

Allen: Almost undoubtedly. In recent years, we have shifted from coordinating all the vulnerabilities for the entire Internet to shifting down towards taking the big ones to taking the ones that are multiparty. The Internet has grown a lot faster than CERT has. Our group is about the same size that it was back in the '90s, different people, of course. There is still a need for some degree of coordinated disclosure in which CERT gets involved from time to time. Our guide came up because we realized that more people were needing to do disclosure and coordinated disclosures. It is not just software vendors. It also is government agencies and any

manufacturer now that has some smart component or IOT [Internet of Things] device, car manufacturers...

David: It plays a role in bug bounty brokers too, like the HackerOnes of the world.

Allen: There are people who provide those services now, too. They are all getting into this disclosure process. We realized that there are a lot of people that could benefit from the advice we have, just based on our experience. The guide actually started as a bunch of the vulnerability analysts at CERT sitting around the table for a few hours. I basically said, *OK, let us just talk about everything that goes wrong in coordinated-disclosure land*. We sat there for a few hours and I just took pages and pages of notes on, *This goes wrong*, and *That goes wrong*, and *There are all these problems that come up*. Then that sort of served as the basis of, *Well, how can we help people avoid those problems when they come up*? And they have happened enough that we have seen it repeatedly. So let's tell people, *You should expect to run into these problems, and here is what you do about those problems when they come up*. I think you have had some experience with some safety-related product vendors who have kind of gone through this process of coming up to speed on things.

David: Right, yes. I mean, doing vulnerability research, you find vulnerabilities and you report them to the vendor. Sometimes it goes very smoothly, but sometimes it also can go a bit poorly. Generally, that stems from mismatched expectations about the process. As a researcher, you come with the expectation that a particular vulnerability may be assigned a CVE [Common Vulnerabilities and Exposures] and disclosed eventually, after being patched, of course. But the vendor may come with different expectations or a misunderstanding of the standard operating procedures for doing coordinated vulnerability disclosure.

I think that was part of the motivation of the guide was that you can point to something that can help vendors, but also researchers, come to the table with sort of a common understanding of what the process and terminology is going to look like. So, we have had instances where vendors have pushed back a lot, and being able to cite the CVD guide as a resource for how the process should play out has been very helpful.

Allen: We have also seen it play out in some strange ways or unexpected ways, I guess, not necessarily strange. Back about a year ago in 2018 when Meltdown and Spectre, when those vulnerabilities came out over the holiday season, there was a lot of concern, all the way up into the government about how those vulnerabilities were communicated to the vendors and to parties in critical infrastructure and the government and all sorts of stakeholders, to the point that there were Congressional hearings about this. Intel and Apple and Google and Microsoft and Arm—and I am forgetting a couple of other vendors—Congress inquired them about how they handled that disclosure. In their responses, some of those vendors actually cited our guide as, *Well, this is*

how we do this. It was a nice endorsement for us that we had gotten the attention of those vendors, and that they had noticed that this is a good framework for which to do this.

Later on in that year, in 2018, there were Congressional hearings or Senate hearings at which Art Manion, our colleague, testified regarding the disclosure of Meltdown and Spectre, and in particular, coordinated disclosure and how that can help protect critical infrastructures in the nation and around the world. As a result of that, we actually got feedback from the House and Senate. The joint House and Senate committees that had sent out those inquiries and also held the hearings, providing us with feedback on our guide. Actually in the latest version of the guide, there is a section in there that is prompted by Congressional staffers and Senate staffers, providing us that feedback from their committees. It was not quite the audience we thought we were going for with this guide, but it was neat to be able to interact at that level of government, where they are telling us, *We want this guide to say this*. We had some back and forth, and we made some updates based on that feedback.

David: That is an important point is that the guide is a living document. We are incorporating feedback from others into it as we receive it and are tweaking it for different processes. Let's see, it is a wiki now, is [that] the best place for people to find it?

Allen: The original version of the guide we put out in 2017 was a PDF file, just a giant report. I think it came in at over 120 pages. As a lot of folks know, 120 pages is a lot to get through. We heard that, but we also realized that the reason it was 120 pages was because we had a hard time saying it in fewer words. Because there's a lot to...

David: There are a lot of corner cases in this process.

Allen: Yes. Again, because it originated as a, *Enumerate everything that goes wrong and then try to provide advice for how to fix that*, we had a hard time collapsing it down. But what we did was we turned it into a wiki that we can now edit and revise as needed. So that was the first major improvement, just getting the content out of the static PDF file into a wiki that we can dynamically edit.

David: It also pares down the size shock or length shock of what you are looking at, right? You are not looking at a 120-page document anymore, you are looking at one specific wiki page, which details one aspect of it. The executive summary is a page in the wiki, I believe so.

Allen: In fact, you can go print out the executive summary just by itself and start with that. It also now links into all of the other sections, so you can hop directly to the parts that you need.

David: That is a great starting point for vendors and researchers if they just want to figure out the gist of it and then drill down to the sections that they may be having problems with, which includes a troubleshooting guide too, I believe.

Allen: Right. One of the things we noticed was… We follow a bunch of security researchers on Twitter, and occasionally they will comment about, *I reported this vulnerability to this vendor.* And they are having some problem with the coordination or problems with getting the vendor to pay attention to them, or whatever the issues are. Just from watching that, we realized, *Oh, there are a lot of edge cases that we actually did not capture*. We captured all of the problems that occur from a coordinator's perspective, but we didn't really have the researcher's perspective quite as fleshed out, because we are not really playing in the researcher-to-vendor space directly so much. We are more in the coordinator space. But from observing that, we realized, *Oh, there are more things we could put in here*.

David: I just want to interject here. Generally, when we do interact as the researcher when we discover something and we are talking to a vendor, we have the benefit of being at CERT to back us up when we are talking to vendors, which is a good token to have. A lot of researchers do not have that. They are just a lonely researcher that does not have an organization backing them, which makes it more difficult for them in some instances. I think we even mention in the guide in the troubleshooting section, *If you can't contact the vendor or the vendor dismisses the vulnerability and says it is not a vulnerability and that they are not going to do anything or address it, then sometimes public disclosure [is an option], even without the vendor's coordination necessarily, if you have done a good-faith effort to tell them about the vulnerability*. That is one from out of the guide. Which is a bit more of unilateral vulnerability disclosure than coordinated. But, it is one aspect of the troubleshooting of all of this that when you are optimizing for societal good, sometimes that is the answer.

Allen: The troubleshooting guide, in particular, is just a small section within the overall guide. But to me I think it is probably the highest impact change that we have made to the guide since its original publication. The idea from that really came from, if you buy an appliance like a washing machine or something, and you look in the manual, in the back there is usually a table that says, *If you have this problem, then do this*. If the washer is beeping, and there is a certain code on the display, it means that the load is unbalanced. A different code means that the belt needs to be replaced or whatever. There are a lot of different troubleshooting steps. But it is really problem-oriented where, *You have this problem. Here is how to recognize whether that is really the problem you have, and then what should you do about it*.

So, we built this table that starts off with a brief description of the problem. It also explains or includes which roles within the coordinated disclosure are likely to experience this problem. Whether that is the researcher, the reporter, the vendor, or coordinator, sometimes it is multiple

of those that might be affected. Also, which phase of the coordinated disclosure does that tend to occur in, whether it is from discovery through initial validation and triage of the report all the way through to getting the patches produced to deploying them or communicating the need for that deployment. There is a long list of issues we have seen, some of which are things that we thought of after we published the guide originally. Some of them are clarifications of things that we said in the guide. Based on observing people's reaction to the guide, we were able to say, *We didn't quite say that. People are interpreting it slightly differently than what we intended. So here is a clarified interpretation of it*. Sometimes it was just based on the Twitter threads where someone was complaining about something, and we are like, *Yes, we have something to say about that*. We just added those in as well.

David: How do people provide feedback? Do they tweet at you?

Allen: Now that it is a wiki, we also have links on every page. In the sidebar, there is a link that says, *Suggest a change*, that leads to a GitHub project. The project itself does not have any content in it. The content is all on the wiki. But you can submit issues directly in that. And we can have conversations in public about, you know, here is what we think the guide should say. We can go back and forth on those issues.

David: That wiki, we will have links to it in the transcript of the podcast. But that wiki is vuls.cert.org?

Allen: Yes, vuls.cert.org [https://vuls.cert.org/confluence/] is the general wiki. The CVD guide is a section of that wiki. If you go to the vuls.cert.org homepage, there is a CERT guide to coordinated disclosure link from that page, which will take you into the guide from there. There is some other advice, like how to report vuls to CERT and some specific things for us interacting with researchers and vendors as well.

David: If I can plug the wiki, the vuls.cert.org wiki has other information, such as a table for speculative-execution vulnerabilities and a list of mitigations and alerts and stuff. As the list of those grows, we have aggregated information about that class of vulnerabilities on vuls.cert.org. So just I am going to plug that real quick.

Allen: A couple other changes we made to the guide after we got it into the wiki as well I wanted to mention, the new version also has information on how to find vendor contacts. A lot of folks reported, if the vendor doesn't already have information posted—which we recommend that vendors do—how do you find vendor contacts? We have done everything from looking them up, Googling for security@ or other email addresses, looking for the security.txt file, which is based on an RFC recommendation. But we have also done things like find the corporate officers in LexisNexis and then figure out who is likely to be a technical officer of the company and then

send them registered mail. Because sometimes that is how you have to get the message through because you cannot find a good point of contact. It depends on how much effort you are willing to put into it. One of the things that is also in this troubleshooting guide is, no researcher is obligated to do any of this. If all you want do is just get it fixed, send it in the best way you know how, make a good effort, and then you can walk away from it, you can publish it yourself. Hopefully you have notified the vendor or made a good-faith attempt to notify the vendor. You can contact us and maybe we can help you find a contact.

There are other organizations and groups around the world that do coordinated disclosure as well. Lots of national CSIRTs [Computer Security Incident Response teams] are starting to build their capabilities there, so you can report to them, depending on which country you are in and your affiliations there. To that end, we also included a little bit more explanation of when and why you might engage a coordinator, as opposed to just reporting to a vendor if you are a researcher.

David: That is a good question. If I am a researcher and I find a vulnerability, why should I come to a coordinator such as CERT as opposed to just going straight to the vendor myself?

Allen: If the vendor has available and easy-to-find contact information, and they have a reputation for treating researchers well, and basically, they appear to be willing to receive those reports, you probably do not need us. You can report the vuls directly to the vendors. Most big vendors are pretty good about this at this point, at least traditional software vendors.

David: The industry is maturing, slowly.

Allen: Lots of cloud vendors and service providers also are pretty good at recognizing the need for this. Where we have seen it more recently, our focus has shifted into the problem cases. The problem cases tend to arise not so much on traditional IT vendors but more in the companies that used to build mechanical products, durable goods, or those sorts of things, and now they have an IOT component to their thing, so it is a refrigerator or a car or a washing machine...or an industrial-control system or a medical device or an airplane.

David: To be clear, it is not all of these companies. Some of them are well versed in coordinated vulnerability disclosure. But especially the new ones, they did not even live that much in computer land and they have a couple of embedded engineers. Now all of a sudden, they have to swim in this pool of coordinated vulnerability disclosure.

Allen: And a few years ago, those companies were kind of surprised to find out that they were computer companies or they were software companies. I do not think that many of them carry that illusion anymore that they are not. But they also still have not really ramped up quite as

much in those industries as like the IT industry has, to recognize the need to be able to receive reports and deal with those things.

So, if a researcher cannot find a contact, or if they have made contact but they have gotten a really hostile response or they have gotten some negative response from the vendor, we will help, and lots of other coordinators will often help smooth that over and can act as a neutral third party to say, *Is it really a vulnerability? Is it important*? Some of that helps with taking some of the weight off the researcher to justify to the vendor, *No, I am not really trying to hack your systems. I am not trying to destroy you. I am just trying to point out that, Hey, you have a problem. It is almost the equivalent of like, Your shoes are untied, and I thought you might like to know*. We can help them through that, and some of that is also encapsulated in the guide. We can contact the vendor and say, *This person is trying to do this thing with you, and here is a guide that explains what they are trying to do and why you might want to treat them nicely because they are actually your friend, they are not your enemy*.

David: I think another category of that, of a time when you might want to engage a coordinator, is when you find a vulnerability in some component that is used in thousands of products or something. So all of the sudden the workload for you as a researcher goes from a one-to-one conversation with a vendor to, *Oh, I have found something in a library or some subcomponent of something that is used across the industry*. Ideally the person that wrote that library would be the one to step in and coordinate it. But even the workload for that can be tremendous, so that may be a time when people would want to engage a coordinator.

Allen: Vulnerabilities can occur anywhere along the supply chain from the library standpoint, and also even in the hardware-component standpoint. Meltdown and Spectre are a classic example. In fact, they are probably the most prominent example that I can think of a hardware problem where, yes, there is really one vendor or a couple of vendors—the CPU manufacturer vendors—are the ones that are best placed to actually fix the problem. But CPUs and operating systems intermix so much that there are things you can do in the operating system that can address problems in the CPU and vice versa.

In situations like that, you wind up with, it is that multi-party coordination process. That is really where the multi-party coordination thing got a lot of attention because a lot of those vendors were kind of new to that process. The Googles and Microsofts and Apples were mostly used to fixing their own problems in their own software. But when it comes to, *Oh, I know about a problem, and it affects me, but it also affects you, and we both need to do something to fix it. Oh, there are also 20 other vendors out there that also need to do something.*

David: Or the least-cost avoider fix could be an OS manufacturer as opposed to a hardware manufacturer.

Allen: Then some of those changes also affect downstream now. As a result of a hardware problem if you change something that happens in the OS, and now an API [application programming interface] changes that your downstream application vendors were using, you have to coordinate with them, too. That supply-chain process or supply chain as a way of thinking about where the coordination goes becomes really important. Supply chains are not easily mapped, and that gets into a lot of complexity in terms of just how software is made.

David: These cases can get incredibly complex. We have seen a couple recently, such as the speculative-execution set of vulnerabilities.

Allen: That is not new. You know, we had a situation...

David: Well, the Kaminsky vul in 2009.

Allen: 2008/2009, there was a DNS vul.. There was an SNMP vulnerability back in 2001 or 2002. At that time, that was the largest vulnerability we had ever coordinated in terms of effort. We had 8,000 or 10,000 email messages back and forth bet.ween us and all of the vendors. That was one of those things where every network vendor, every operating system vendor, anybody who made anything that talked TCP/IP probably had an SNMP client that they needed to fix.

David: The protocol for spec vulnerabilities is always tricky.

Allen: Yes, the deeper it is embedded in the system, the harder it is to get it fixed because there is a lot more coordination because lots of people have to do something.

David: Switching gears a little bit. We have recently published vul-disclosure policy templates. Do you want to talk briefly about what a vul-disclosure policy template is and then how people could use it in their companies or as researchers?

Allen: We have had to review a lot of organizations' vulnerability-disclosure policies. Sometimes that is because our government sponsors have asked us to review their own agencies' policies. Sometimes that is because we have been asked to develop policies or develop text that can go into their policies. We have also been involved in multi-stakeholder processes out of the NTIA [National Telecommunications and Information Administration, US Department of Commerce] groups in vulnerability disclosure a few years ago, as well as some industry standards organizations. There is some work in ISO for vulnerability disclosure [see here and here]. There have also been some other collaborative efforts about vulnerability disclosure.

What we have realized is most of these policies are saying almost the same thing. They are sometimes using different words. Some policies are addressing some aspects, but they are missing other things that we think are important. So, we put out a set of templates that are really intended for… It is not that you can just take this document and turn it into your policy, it is that

this is a document that has a list of statements that you might want to include. You should evaluate them individually and pick the ones you want, pick the ones that make sense for you.

David: Can you give me an example of some of those?

Allen: If you are a researcher and you are reporting a vulnerability to an organization, you probably would prefer it if the organization didn't require you to be a customer and have a customer agreement before you could report things, because I am just a random guy off the street that noticed something, and I want to tell you about it. I do not have a customer ID. So, there is a statement in there to that effect, that organizations should not require researchers to be customers to report vulnerabilities. There are things like, researchers should keep their findings private, for whatever duration it takes, either some specific timeframe, like 45 days or 90 days, depending on your policy. Or, maybe it is until the problem is fixed and the fix can be confirmed.

David: So, people are going to have to tweak it for their individual solution.

Allen: Yes, we wrote it with RFC-style language with, you know, the *musts* and the *shoulds* and the *shalls* and whatever, mostly because that lets us be very succinct and clear. These are single bullet points where each one makes exactly one statement about an expectation that either a researcher or an organization, a vendor, should have or may have of the other party. The idea for using this would be, you go through that list, pick out the ones that make sense to you. You can tweak the wording. If ours says *must* and you think it should be a *may*, fine, do that. But just try to be a little more consistent in what things you are addressing.

David: It also helps vendors and researchers be on the same page as far as what is going to happen, right?

Allen: Most of the problems we run into with vulnerability disclosure tend to be because of mismatched expectations. So, this is really about helping to nail down those expectations of, *Researchers can expect organizations to behave this way, and organizations can expect researchers to behave in a certain way*. They can articulate that in somewhat of a common language using the templates we have put out. The whole point of that is, currently there is a request out from Office of Management and Budget [Cyber and Infrastructure Security Agency] for a potential binding operational directive to go to all the government agencies saying that government agencies will have a vulnerability-disclosure policy, which means there are going to be a lot of agencies out there that need to write policy.

David: Each agency will have its own?

Allen: Each agency will have its own policy. They are all going to need to do that in 2020 if that directive becomes actual fact. So, we thought it would be a good idea given that we have done

this enough times, we should just put some words out there that can serve as a starting point for those agencies, but also for vendors, companies, or service providers, or for that matter, researchers. I mean, a researcher may want to have an individual disclosure policy that, *Here is how you can expect me to behave*, and they can be declarative about that. Ultimately, being declarative about this is good because it sets the expectations and then avoids the uncertainty. The place where problems arise is when there is uncertainty in the process.

David: So, Allen, thank you for taking the time to talk with us today. The links for everything we talked about will be in the transcript. Thanks again for joining us.

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.