VINCE: A Software Vulnerability Coordination Platform

Featuring Art Manion and Emily Sarneso as Interviewed by Allen Householder

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.

Allen Householder: Hi, my name is Allen Householder. I am the technical lead for threat ecosystem analysis in the SEI's CERT Division. Today, I am joined by Art Manion and Emily Sarneso. Art is the technical manager for Vulnerability Analysis in the SEI CERT Division, and Emily is an application developer, also in the SEI CERT Division.

Today we are here to talk about the VINCE, Vulnerability Coordination Platform. So, this is the latest of several podcasts that our team has been recording on vulnerability coordination and reporting. We will also include links to the previously published podcasts in our transcript. Emily, Art, welcome.

Emily Sarneso: Thanks, Allen.

Art Manion: Thanks, Allen. Great to be here.

Allen: Let's start off by having you tell us a little bit about yourselves. What brought you to the SEI, and what do you do here on a daily basis?

Emily: I came to the SEI from grad school. I was at the Pitt iSchool previously, and I am an SFS (Scholarship for Service) student. I was in the Information Security Program at the iSchool at Pitt and did an internship with the CERT/CC [the SEI's CERT Coordination Center]. Upon graduating, I went directly to CERT in the Network Situational Awareness team, where I was an application developer. I was working on network-monitoring tools, such as YAF (Yet Another Flowmeter) and fixbuf and super_mediator. I joined the vulnerability team about three years ago, and have worked on a few things before helping out with the Vulnerability Coordination Team. I am the lead primary developer of VINCE, and that is where I am at today.

Art: Hi, so this is Art. I have been with the SEI for 20 years now. I have been almost the entire time hoping and wishing for better vulnerability-handling tools. Finally, in the past couple of years, Emily has come on to the team, and as she mentioned, is the lead developer for the VINCE project. It has been very exciting after these 18 to 20 years to finally be building some new tools for our work. I am the technical manager of the Vulnerability Analysis Team, a small team. We are 13 people, I think, and a core part of our work, not the only part but a core part of it, is coordinated vulnerability disclosure. We offer a service to the public to take in reports about vulnerabilities. We select which ones we think we can add value to and make a difference on, and we coordinate those. That entire process we will talk about more during the podcast, but again, VINCE is key here in being our new modern tool to help us with our coordination work.

Allen: I am Allen Householder. As I mentioned, I am the threat ecosystem analysis technical lead. My background at CERT goes back almost as far as Art, although I did leave for a few years there, 2007 to 2010, and then came back. So, I have been back for about 10 years. More recently, I have been working on vulnerability-discovery tools and fuzzing tools, and then also doing some data analysis and threat analysis. Part of my role too here has been to get to the process of coordinated disclosure. So, I was a coauthor of the CERT Guide to Coordinated Vulnerability Disclosure that we put out a few years ago. There is another podcast on that. We will have links in the description box for that. Up until today, pretty much what I have been working on for the last decade or so has been vulnerability-related discovery or coordination or things in that general realm.

I would like to start by talking a little bit about vulnerability management and the genesis for this work. I realize up until just recently, our coordination work has mostly been sort of a hub-and-spoke approach where CERT/CC has acted as the hub of a coordination process between researchers and vendors, fielding those reports, and managing that process. I understand that VINCE has changed that quite a bit. Can we start by talking a little bit about what has changed with the process and what VINCE is trying to address?

Emily: Sure. So, when I joined the team about three years ago, I sat down with all of the vulnerability coordinators, and I had them break down how they currently did their work. We started with just a blank whiteboard and all of the different tools and processes and workflows that they currently use in vulnerability coordination. What I realized was that in order for me to tackle any one part of that, we needed to look at the whole process and decide if we can change the way that we do our work. Eventually, the team agreed. It was a slow start. We started replacing little pieces of the tools, and then realized, *You know what? Let's just scrap it all, and we are going to move to something where it is more coordinated and collaborative, and rather than us being the middleman for all of these processes*. So, that is the key to VINCE, is that we are trying to bring everybody into a common room. We call it our "case discussion," and we give everybody the information that we were given. Usually a reporter provides information to us about a vulnerability, and we provide that information to all of the different people that we think may be affected by that vulnerability, and we discuss that. We figure out what is the best way forward. We talk about disclosure plans. We talk about remediation plans. We talk about how we are going to disclose it to the public. We hope that everybody gets involved—the vendors and the reporter and the coordinator—that we all communicate and collaboratively come up with a plan for disclosing the vulnerability and giving that information to the public.

So, that is the whole idea behind VINCE. It is a web-based software. People sign up for accounts. They log in, and they have access to all the information that they need, all the information that we were given. And they provide us with more information, so that we can provide that out to different vendors and to the public eventually.

Art: Yes, thanks, Emily. There is a key piece here that Emily highlighted early on. Previously, using email, PGP-encrypted email, to speak with many, many different vendors on a case, we were at the center of this hub-and-spoke model, and, in many cases, the CERT Coordination Center was delaying—causing delays—and blocking communication. One of the philosophies behind VINCE is that researchers, reporters of vulnerabilities, and the vendors, and the coordinators—ourselves in this case—are all able to speak in a single-channel case chat that Emily described. That is one of the ways we are trying to unblock progress in the coordination process. This becomes very important when we talk about multi-vendor or multi-party coordinated disclosure.

There are many software developers and development organizations today that have software vulnerability response and PSIRT [product incident response teams] capabilities. If you find a vulnerability, you can go straight to them and coordinate with them. You do not need a third-party coordinator like the CERT/CC here. But, we often find ourselves involved these days with cases where it is hard to reach a vendor, or a vendor and a reporter might disagree, or in many cases, when there are many vendors, multiple vendors involved. That is really where VINCE shines. It is designed for this multi-party coordination case, which is much more complicated than filing a bug in a bug tracker between two different parties.

Allen: So, Art, I have a clarification to ask of you here. You mentioned “reporters.” I think we all kind of understand what vendors are, but I have heard “reporters.” Some folks talk about “researchers,” other documents talk about “finders.” Can you help explain what those words mean? Are they all the same thing, or something different?

Art: Sure, and of course we can point you to our master’s thesis on the topic, the CERT Guide to Coordinated Vulnerability Disclosure, to let you read some of the details of the definitions. But we, for many years here, have talked about “reporter.” And very specifically, this is the person or organization who told us about the vulnerability. That person may be the finder. They might call themselves a "researcher." Very commonly in the community, we hear the word, "researcher." But "reporter" was just a more specific term. It implied less. It does not imply that this is the first person to find it, the only person to find it, that the person or organization considered themselves a researcher or not. So, when the CERT/CC says, "reporter," and VINCE says, "reporter," that is very literally the person or organization who reported the vulnerability. You can pretty easily substitute the word "researcher" or "finder" if that helps you think about it, but that is our specific term.

Allen: I think the term "coordinator" is usually us in the VINCE context, but are there other coordinators out there?

Art: Yes, thankfully, there are. So, we do this operational coordination work. We offer this service. But we are trying—it has been a long, strategic slog—but we are trying to put ourselves out of this operational coordination business really. There are other coordinators in the world. There are other coordinators in the United States. Many vendor PSIRT teams act as coordinators from time to time. We work closely with the Department of Homeland Security [DHS] CISA [Cybersecurity and Infrastructure Security Agency]. They do coordination. We work with our counterparts in at least Japan, Finland, the Netherlands, so it is possible…a "coordinator" is a generic term for the third-party organization or role that helps coordinate the vulnerability response. Allen mentioned "vendor" earlier, but just to be careful about it, "vendor" is code for "software development organization." When we say, "vendor," we absolutely include, for instance, open-source maintainers. It is the producer of the software, and it is the organization or person who is involved in producing a fix. Again, "vendor" is the broad term, but it absolutely includes open-source development.

Allen: And also, increasingly includes traditional manufacturers, too, as they involve software into their systems with the products that they make, right?

Art: Absolutely. We are finding, it is not our phrase, but I have heard a lot of it. The phrase I hear is, *Everyone is a vendor now*. Almost any organization has a website, possibly a mobile app, and to Allen's point, right. Companies who have been around many, many years, decades, and are very well established in perhaps a physical product-manufacturing ecosystem or sector are now also software vendors because their physical product now has a computer attached to it, talks to the Internet, talks to a cloud service, part of the Internet of Things [IoT] world. So that is a whole new sort of spectrum, a whole new area of relatively new software vendors coming to the coordinated-disclosure discussion.

Allen: Thanks. So, I would like to shift gears to talking actually about VINCE at this point. How does it work? What does it do? So, to start with an easy question, Emily, where did the name "VINCE" come from?

Emily: I get this question all the time. I had previously developed a tool called "MARTIE." MARTIE was a malware analysis platform tool. To stay within that naming scheme, which was just people's names, we are trying to "backronym" a name for this new thing I was developing, which was the vulnerability coordination platform. So, there are very few names that actually start with V and kind of are short enough to backronym. So that is where VINCE came from. It is the Vulnerability Information Coordination Environment, and we did a little bit of fuzzing there with the INformation, IN, which…so people always ask me, *Why didn't you call it "VICE"?* And that would have been a fine name had I not had the limitations that I set upon myself to find a name. So, VINCE, this is where the fun starts, really, talking about how VINCE works. And VINCE is a web-based software. It is primarily developed in Python using Django. It is hosted in AWS [Amazon Web Services]. We use cloud-formation templates to bring up the system. There are actually three different systems in VINCE: One part is our public website, which is kb.cert.org, where you can find all of the information about VINCE.

We have another part of the software where it is our collaboration piece, where vendors are able to log in and communicate with us, and reporters. And then there is our tracking piece, where the coordinators log in and do vulnerability-coordination work. They are kind of the people that are behind the scenes that are giving people access to cases, developing the cases, doing the required analysis on cases. So, it all starts with the coordination piece. When a reporter comes to us with a new report about a vulnerability, our vulnerability coordinators take that information. They might do a little research about that vulnerability, confirm that it is, in fact, a vulnerability, and then they go out and they try to decide who is affected by that vulnerability. So, part of that tool is our contact-management software, where it basically just keeps track of all of the different vendors that we coordinate with regularly. We have a large amount, about 2,600 I think, of vendors that we have communicated with in the past. So, we use the VINCE software to contact those vendors and ask them to come to the case discussion, which is the collaboration part of VINCE.

Once the vendor gets access to the software, they can log in. It is more like a message board. People can communicate. They look at the original report, the report that we received, with all the details about the vulnerability and how they can reproduce it. Then they can view the vulnerabilities that we have identified. They may or may not have a CVE [Common Vulnerabilities and Exposures] ID at that point, and then they will eventually get to look and critique our vulnerability note, which is typically where we publish our vulnerability notes on kb.cert.org, so they can see drafts of that publication as we move forward. And they can communicate with the reporter as well.

Oftentimes, the vendor has questions about how to reproduce the problem or the bug in the software. So, having the reporter in that case discussion really takes us out of being the middleman because, often, before we had VINCE, the vendor would come back to us and say, you know, *We need more information about this problem*. And then we would have to forward that to the reporter. The reporter would comment back to us. We would send that to the vendor. Now everybody is in the same room, and they can talk back and forth. They can share files. They can share information. We have already had a couple of cases that we have handled this way, and it has really been great, and it has moved things along much faster than us having to be in the middle of that whole process.

There is also an API [application programming interface] that we have developed for vendors that do not want yet another account somewhere on the Internet. They do have to have an account, but then they do not have to log in and get the information. They can just use our API to pull that information into their own systems. Most times, PSIRTs for organizations have their own tracking systems, which is how they track bugs in their software. So this way, they can pull that information right into their own tracking system and be able to move that along internally.

Allen: So, I know at the beginning of this project we had a lot of discussion about whether or not we needed to do custom development or if we could just use some off-the-shelf software, like a ticket-tracking system, for this coordination system. And we eventually landed on needing to do custom development. Could you explain a little bit why that was the conclusion?

Emily: We originally did not want to create a custom ticketing platform. We looked at a few different ticketing platforms, such as Jira and TheHive. MISP and TheHive was a great tool, and we really wanted to be able to use something like that. The issue was that we have a very specific use case for multi-party vulnerability coordination. And being able to manage tickets per vendor was very difficult in those other ticketing systems. We need to have almost like a thread for each vendor and be able to manage the information that we receive per vendor, and then be able to search on those kinds of things, and then publish on those kinds of things. It was really just easier to create our own at that point. Fortunately, there are quite a few systems out there that we were able to kind of start from. We were not starting from scratch, and we were able to then take our specific use cases and then develop those into that.

Allen: So, Art, we have talked about the technical underpinnings of what VINCE is, but who would you say the intended audience is for VINCE?

Art: Well, clearly, it is the audience that exists for this coordination work in the first place. So, we have talked a little bit about some of the roles, but number one audience I would say is probably the vendor community. So, these are, again, the folks who are developing software, responsible for fixing it or maintaining it. It is fairly straightforward logic. If a vendor is not going to fix the software because they are not aware of the problem, we are not going to get any further with CVEs and patches and securing our systems.

So, it is imperative that vendors find out about these vulnerability reports, so, certainly the vendor community, also, the reporter, or researcher, or finder community. We are all kind of depending on the beneficence of some researchers when they find a security defect to report it to a vendor or to a coordinator. Security researchers have options. They do not have to report to a vendor or a coordinator. So those are our two probably largest external audiences in terms of getting the coordination work done. Emily mentioned vulnerability notes previously, and that is our public advisory about the vulnerability. So ultimately, the reason we think this coordination work is valuable is that the end result is information that system owners and operators and users can then use to secure their systems, apply updates and patches, take mitigation steps, assess their risks more appropriately. While the end audience of vulnerability management is folks who administer and use and operate systems, coordination parts that happen before the public disclosure, for the most part, are really the vendor community and the researcher community, or reporter community.

Allen: All right, so a question for both of you. How long has VINCE been in use? And what have we learned so far from having used it so far?

Emily: VINCE has been in production for over a month now [as of June 2020]. We have over 200 users at this point, internal and external, all over the world. We have learned quite a few things in this first month. We have tweaked a lot of things. It is hard to develop a system—this is me personally talking—it is hard to develop a system where people have been using the old system for over 20 years. So, they were very...

Allen: Acclimated.

Emily: Yes, very acclimated to using a particular system and the workflows that came along with that particular system. And so, now we have really changed this whole workflow for all of the people that are doing the vulnerability-coordination work. So, we have really kind of tweaked it as we have [gone]. We realized that things need to evolve as we are moving forward with this new system. So, I have about 250 Jira tickets waiting for me, and we plan to keep evolving it as we move forward. But we have had great vendor feedback so far. People are excited about using it. They find it…it is definitely, I think, easier to use than PGP email, which I refuse to use, so having to not deal with that is one perk of VINCE. They can get their information faster. We have had a couple of case discussions where the reporter talked directly to a vendor and was able to communicate back and forth about a particular problem. I think one of the vendors was from halfway across the world. So, you know, it was interesting to see all of the different people using VINCE so far.

We have had a couple people use our API and are providing feedback for us. So, that is always great. It is great as a developer to get instant adoption, I guess, in this case where we have, you know, a lot of people using this system right off the bat. We have been playing with VINCE for most of this year. We presented it at RSA in February to a group of vendors that meet with us at our RSA vendor meeting every year. So, we were able to demo the system and provide test accounts for particular vendors. We have gotten a lot of feedback from them over the past few months.

We also did some usability training earlier this year with vendors and reporters for our system, so that was a huge help in getting initial feedback for the VINCE system. Additionally, we talked about VINCE at a PCERT technical exchange in March, and again got more people enlisted and got feedback from them. So far, the feedback has been great. Originally, people were hesitant to come to a communal discussion, especially with the reporter being in the room. As they started to look at VINCE and realize that it does actually speed things up quite a bit, and most reporters just really want what is best for everyone, which is they want the vulnerability fixed in a timely fashion. So knowing that the intentions of the reporter are typically good, everyone seems willing to sign up and move forward in this new way of vulnerability coordination.

Allen: So, Art, I know you are very often out at conferences and meetings, and flying all over the planet talking to people about vulnerability coordination and how awesome CERT/CC is, and how good we are at it, and how much we love doing this stuff. How much pushback did you get when you were having those conversations, either early on, and do those concerns still seem to be present, or have we alleviated those concerns by the system we have delivered? Or what is your impression of sort of the vendor community's opinion of the utility of something like this?

Art: Well, I think to some extent that remains to be seen. Emily mentioned around 200 users, and I am really making a pretty rough estimate here, but my guess is that is 10 percent or so of our current contact lists. But what is really interesting—we have hit this point a couple of times, but I think it really bears repeating—this philosophy of assuming that people have good intentions, right, the researchers and the reporters are trying to get things fixed by coming to the CERT Coordination Center or going to the vendor. And the vendors are planning to fix things and do want to fix things. We have looked at this problem from being in the middle. We have seen from all two or three sides of it for decades now, and what is really nice about VINCE is the technology and the design of VINCE actually implement some of this philosophy. So, we have heard stories of reporters who are uncomfortable being known to vendors. We have heard vendors who are uncomfortable knowing that a reporter of the vulnerability is sort of in the room, in the chat space, at the same time. But so far, we have chosen to proceed with creating this room and this space. Early to tell, no final results in yet, but we plan to continue this direction. We expect there will be disagreements from time to time, and we will address those issues as they come up. But there is an even higher strategy, or an even broader strategy here of we are trying to get this coordination and the disclosure process to be sort of a normal, mundane, boring thing. Software has vulnerabilities. People are going to find the vulnerabilities. We believe the least cost to everyone, and that includes the greater good and the social cost, is that there is a report, that the vendors find out, that there is a bit of private embargo time followed by public disclosure and fixes. It is not going to stop happening, it is not going to go away. So, can we make this process as smooth as possible? Again, VINCE is designed and implemented to do that, and that is the really great thing to me, that we get our philosophy turned into code in the VINCE platform.

Allen: That all sounds interesting and exciting actually. So, for a software vendor or a cybersecurity researcher, where do I start? How do I use VINCE? How do I access the platform? What do I need?

Emily: So, you can just simply go to our website that has been our URL for 20-plus years, kb.cert.org, and there is now a VINCE tab at the top. You can create an account there. Once you have created an account, you will be asked to enable two-factor authentication. Then if you are part of our contact-management world, we already know you, and you know us maybe. You are automatically accepted into the VINCE world.

If you are new to us, we will look at your account and probably accept it within a few hours. But you can always submit a vulnerability report to us with or without a VINCE account. We prefer you have a VINCE account before you submit a vulnerability to us, that way we can invite you to that case discussion when we have it, and it is much easier to do that if you create an account first. But once you create that account, we will accept you into VINCE, and you will be able to communicate with us about any reports you make. Or if you were part of a vendor organization and you are affected by a vulnerability that was reported to us, we will be able to discuss it with you there.

Art: As Emily has said in some detail, yes, VINCE is in production. You can sign up for an account now. A couple of things I would like to add. A very soft rollout for us. We are not yet requiring anyone to use VINCE in order to receive reports from us. But as time goes on, we will be steering away from our previous transport mechanism, which is, again, PGP email. So, we would definitely encourage our vendor community to sign up for VINCE, check it out, see how it is going to work for you. Consider the API if you would prefer to have your system talk to our system as opposed to depending on human analysts or on everyone’s individual new web-based platforms.

And to clarify something, we accept anonymous reports, and that is still true with VINCE. A reporter has a slightly better experience, potentially, if they have an account with us. We can add them to a case. They can see what is going on without having to ask us questions through some other channel. But, we continue to support, and have no plans to discontinue support, for receiving anonymous reports. To the extent that we can guarantee anonymity, we are happy to do so.

Allen: And even if a researcher or a reporter just wants to basically drop the report on our lap and leave, they can still do that, too, without even creating a VINCE account, right? They do not have to be anonymous. They can just submit it to us. We still know who they are, but it does not require the creation of an account, right?

Emily: Yes.

Art: We appreciate reports in any form. We prefer higher quality reports, but we appreciate any reports, and we will do our best to handle all of them and make a decision on each one.

Allen: Emily, you mentioned two-factor authentication. Are there special tokens one needs, or is that just app-based, or how does one get the second factor?

Emily: You can use your favorite two-factor app, like Google Authenticator, Duo, Lastpass. There are a few of them out there. You can use one of those, or you can use just SMS-based two-factor authentication, where we will send you a code to a phone number, and you use that code to log in.

Allen: Okay. So, Art, over the last few years, we have continued to refine our work in vulnerability coordination, so what is next for us?

Art: We have discussed some of this throughout the podcast so far, but at a strategic level, I am interested really in this normalization, this making coordination and vulnerability reporting boring, everyday, and common. We do not, as a society and as a country, have the energy and the bandwidth to treat every disclosure like an emergency. We are counting 20,000-plus public vulnerability disclosures per year. That sort of scale has to be automated. That is also a low watermark, by the way. The number is probably much higher. So, we need to reduce friction for this entire process, from the finding of the report to the last moment that the last point of the system is patched, if that ever happens, and make it kind of boring, and not exciting, and not dramatic, and routine, and as efficient as possible. And again, limiting risk and harm and cost to all of the parties involved. The API Emily mentioned might be a specific way to do, help do that within VINCE and across other such platforms. The CVD guide we mentioned earlier is part of our attempt to influence process and standard development globally, to make this process normal and boring. In an operational sense, we want to put ourselves out of business, we want more coordinators. We want coordinators closer to their respective sectors. We want all kinds of vendors, open-source, IoT, traditional compute, traditional IT, physical product and goods vendors who are now also software and computer vendors—all of them to have vulnerability-response and vulnerability-disclosure programs so they can handle these reports themselves, ideally, someday without the need for a single, global, third-party coordinator like us.

Allen: So, Emily or Art, anything else that I did not ask you about today that we should cover before we close?

Art: I think we have covered all the major points and covered a lot of ground. Thanks for interviewing us, Allen. We encourage our vendor and reporter communities to sign up for a VINCE account, to check it out. We are truly testing in production. We would greatly appreciate your feedback. We will eventually be moving to VINCE as our primary way to share vulnerability information.

So, particularly for our vendor community, we recommend signing up for an account. Check out how VINCE is working. Test out the API. We do plan to open-source VINCE at some point, once it is a bit further along in the development process. I really cannot thank Emily enough for her development activities in getting VINCE up and out the door and into production, and I am sure the few remaining months of additional feature requests and bug fixes that we have moving forward.

Allen: Okay, so if folks want to use VINCE, they can go to kb.cert.org and click on the VINCE button up at the top. And Emily and Art, thank you for being here today to talk about this work.

Emily: Thank you, Allen.

Art: Thank you.

Allen: And to our listeners, thanks for joining us today. We will include links in our transcript to all resources mentioned in the podcast. And this podcast should be available at the SEI website at sei.cmu.edu/podcasts, and anywhere else that you get your podcasts, including iTunes, YouTube, Stitcher, and SoundCloud. Thanks.

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.*