#171 - Navigating Software Supply Chain Security (with Cassie Crossley)

[00:00:00]

[00:00:00] **G Mark Hardy:** Hello and welcome to another episode of CISO Tradecraft, the podcast that provides you with the information, knowledge, and wisdom to be a more effective cybersecurity leader. My name is G Mark Hardy. I'm your host for today, and we're going to be talking about software supply chain security with the lady who literally wrote the book on it, Cassie Crossley.

Cassie, welcome to

[00:00:32] **Cassie Crossley:** Thank you so much for having me. I'm so excited to be here.

[00:00:37] **G Mark Hardy:** This is going to be a really good episode. So if you're listening to us for the first time, welcome to CISO Tradecraft. If you're a regular listener, Please make sure, of course, you're subscribing to us on LinkedIn because we do a whole lot more than podcasts and, check out our YouTube channel if you haven't done already and go ahead and subscribe because then you get to see my smiling face along with our beautiful guests when we do these types of shows.

Now, you, as I said, have just written the [00:01:00] book on it. And

[00:01:01] **Cassie Crossley:** Yes, I do. I have it right here

[00:01:03] **G Mark Hardy:** up and say, there it is. Offer supply chain security and O'Reilly. So you've got a bird on it is O'Reilly always has a different animal. And what did you select for the

bird?

[00:01:13] **Cassie Crossley:** it is an Indochinese rollerbird. they actually selected it because the first version had an ant and ants and I don't like creepy crawlies. So I asked them to use something else and they You know, they have identified this beautiful bird, so I was very happy with it.

[00:01:33] **G Mark Hardy:** Alrighty. Sounds good. Well, that sounds like at least it's an improvement over something else. I mean, I'm trying to think like, yeah, I could see Ants with Carpenter, you know, whole sub chain of them bringing stuff and things like that. Well, one of the interesting things about supply chain is we really touch into logistics and that's how a lot of people think about a supply chain.

If I need to buy a product or something like that, it's going to go from here to here to here, and it's eventually going to get [00:02:00] to me. Now, a quote that I remember, I might get it close, but it was General MacArthur. who during World War II was responsible for the island hopping campaign that worked its way across the Pacific.

And when he was dealing with all these issues, because he's more of a strategist and a warfighter and the like, but of course, everybody needs logistics. And here's a quote that said, I don't know what this logistics thing is, but I want more of it. And so we find out then is that supply chains are incredibly valuable.

And your book introduces a concept of software supply chain security. And in there, from my review of it, you summarize infrastructure, security controls, you look at like a secure development life cycle and how that could impact, the types of source code, managing integrity, the SBOM, the Software Bill of Materials, assessing third party suppliers, assessing third party, personnel risks, and like 80 different controls that you could do for software supply chain security.

And I guess you get those away on your

[00:03:00] website as

[00:03:00] **Cassie Crossley:** That's right. the, the book itself is because I spend so much time working with suppliers every day. I'm the vice president for supply chain security at Schneider Electric, and we are a 36 billion euro company headquartered in France. And when you're working in building 15, 000 intelligent products every day.

you really get to see the entire supplier landscape and I was spending so much time, providing links and mentoring these other suppliers and even companies, startup companies, all sorts of different groups on what they need to consider because it's so much more than what people might think. Just even software.

And, you know, in my role where I'm working a lot with our teams on developing Operational technology products. It's very important because most of those are in some ways cyber physical and related to safety that you need to be very diligent about the security posture of the products, [00:04:00] how it's put together.

So I used all of that and leveraged it and, was able to put it down in a book.

[00:04:09] **G Mark Hardy:** Yeah. And that's awesome because a lot of times people gain some expertise or whatever, and then they take it with them when they retire. They're like, man, I wish we had her back. Cause you do this stuff, but you took the time and effort to write a book, which is awesome. for, and I know that at one point in time I had, Bill, on here from NoStarchPress because he deals with a lot of publishing.

And I know you went with O'Reilly, but it's interesting because for people who think they have a book inside them, go look up that episode that we had, on that particular one. I will also put in our show notes, a link to your book so that people can find it on Amazon because it looks really, really helpful.

And the interesting thing is that some people say, well, does it pertain to me? And whether I'm a CISO or a director or a VP or something like that, I buy Microsoft stuff and I run it on Dell [00:05:00] computers. So I have two suppliers and I don't have any supply chain issues. So why am I going to waste my time with this episode?

[00:05:05] **Cassie Crossley:** I

[00:05:05] **G Mark Hardy:** And you're laughing about that. So disamuse me and of

[00:05:09] **Cassie Crossley:** Well, I think that anybody who is in any company knows that they have just alone at least hundreds, if not thousands, of software and digital products in their environment. you can't depend on only a couple suppliers and who are they depending on? we have a product that we did what's called an illumination and that product had only 35 or so direct suppliers, both at the hardware and software, and it expanded out to 4 to 498.

dependencies and upstream suppliers when you think about open source and commercial libraries and all of the different chip manufacturers. So, You're not in this, you know, by yourself. It's not just that supplier and they are working,

I'm sure, especially [00:06:00] the larger companies, just as diligently to understand their suppliers and upstream.

So, it's a very daunting task and you can't just depend on, okay, well, this is covered because they have this certification.

[00:06:17] **G Mark Hardy:** Right. I was looking at my desk for a ordinary pencil. And there's somebody that told me a few years ago, they're going to try to say, Let me make a pencil without using any suppliers. So what do you need? Well, you need a tree, you get the wood, you need graphite, you need paint, apparently something to make the eraser and the metal and the crimper on that.

And I guess over a period of three or four years trying to do all these things, such as, well, how do I get to get the pencil? I got to cut it down. I think it's an octagon. Maybe it's a hexagon. It is a hexagon. But then how do you get this in here? And it turns out we have so many different things that go into that by the time we get to something as simple where you buy.

These things, a whole box of them at the dollar store, which is now a dollar and a quarter store, by the way. [00:07:00] But the thing is, is that the simplest of things have this huge complexity behind it, because all interconnected. And so as a result, when we look at physical things, it's instantiation. We can say this came from there, this came from there, this came from there.

If you've ever built anything, you know that you have all these different supplies. Think about how complex an automobile might be. But software, Because parts can come from literally anywhere in the world and don't have to be physically transported and something that works gets reused and over and over again.

And if you take a look at some of the stuff, even in Windows 11, you pop up some screens when you go into settings or whatever, and you take a look at the IPV4, 48 network, and it looks an awful lot like XP and it might even be Windows 95 and that because it hasn't changed.

And so why change it? So there's a couple of things that

go

into supply

[00:07:47] **Cassie Crossley:** I have an interesting story on that. that's in the book is

[00:07:50] **G Mark Hardy:** Okay, I want to hear

[00:07:51] **Cassie Crossley:** if you have a Windows machine, you have lines of my code because I worked at the company that sold paint. exe to [00:08:00] Microsoft. And they're still using some of that code on there. So that code was written in the late 80s.

So, I mean, they didn't refactor it.

[00:08:10] **G Mark Hardy:** and that is well written code if it's still going 40 years later. And then when we look in the mirror, we go, you know, that's where that gray hair is. You

[00:08:18] **Cassie Crossley:** Right.

So it

lasts forever. I mean, who's going to rewrite an entire thing? They put out the 3D version of paint, but, you know, why touch, you know, that old code?

[00:08:28] **G Mark Hardy:** don't like it. I, I still literally use paint. And the reason being is I'll do a screenshot. Now I can do windows shift S and I can go grab a little cut, but then what? Yeah, and then you don't get that. I just stick it in paint because I know how to do these things. I'm going to add this little caption, put the arrow in there, cut, and yeah, we look at these things and maybe we become a little bit of a Luddite in a way that we say, I'm going to use the old stuff.

Why do something new? But the reality is, if it works, don't fix it. That's kind of the first law of rule mechanics. Of course, the second law of rule mechanics is if you tinker with [00:09:00] it, you'll break it. And so to avoid rule two, we stay with rule one. And also it's faster and cheaper. Now, as we look at the idea of supply chain, and then specifically what you're talking about is supply chain security for software, we find out that even in the physical world, there's concerns about supply chain security.

And probably one of the things that are within memory for a lot of us, maybe not everybody on this call, was Tylenol. And that was also 1980s. what happened back then for those who

don't So the Tylenol story, I did reference that in my book because at that point there had not been regulations on packaging and how to put things together and what was all a part of that process. So just that one. you know, serious episode where several people died because someone tainted the material.

[00:09:50] **Cassie Crossley:** That's no different than a malicious actor going in and modifying code today. They made modifications to a product that went undetected. And kill [00:10:00] people. The same can happen with source code and software today. People can get in, and it can impact utilities and critical infrastructure. And, I mean, we just saw, United Health Group.

Yep. And, you know, there are, practice, there are software code all around right now that are used for life saving and making sure everything goes, and executes properly. And all of that. is we have to assume that there is some integrity and some that it is maintaining what was it's intended to do and that no one's modified it or can impact it and that's why the security aspect of the software supply chain is very important and it's one that isn't static.

it is so dynamic. Things are changing every day. There's new vulnerabilities announced every day. You have to continuously monitor and keep updated. And that's a lot of work. I [00:11:00] mean, it's, how many, almost 30, 000, vulnerabilities noted last year, and who knows, it'll be 40, 50. And some of those, I don't know.

You know, it's not surprising to see half of them be critical or high every day that are being, announced and, and from us, you know, a CISO, especially in this day and age, you have to spend so much time identifying, what are the key? products that you have in your business, because you have to assume there is going to be some level of software supply chain security attack in these products, because the defense that's necessary to, you know, prevent all, you know, external facing threats is more than, you know, you can possibly keep up with.

[00:11:52] **G Mark Hardy:** and we think about it from a security perspective, our old classic one on one definition of confidentiality, integrity, [00:12:00] availability,

really kind of talking about integrity here, right? Because a lot of what we perceive as a risk, let's take ransomware, for example, it's for the most part for years was an availability attack.

More recently we're seeing it as a confidentiality attack, but here we're talking about something entirely different, which is. Not hitting the headlines as much as that's the integrity. And then if we think about it from a software perspective, recent events in the last few years, like SolarWinds or Log4j, and I'm sure we can come up with more examples and we'll come up with more examples where there is situations where vendors will stick something in there.

They'll have a library that OpenSSL kind of comes to mind where it works and it's free. So we slap it in there. So even though I don't use it. I'm sure I got an app that has it. And then what we find out is that we listen to some of these security researchers and say, Hey, there are still so many thousands of systems that are still vulnerable to log4j or still [00:13:00] vulnerable to some particular vulnerability because just because it has a CVE assigned with a high CVSS score doesn't mean that it gets eradicated.

Out of the ecosystem. Now, to a certain extent, it's like a vaccination for some sort of communicable disease. If there's a new problem, of course, we went through a pandemic, but if you take something that says, okay, fine, everybody's potentially vulnerable to measles, pick measles. And well, if all your kids get vaccinated against measles and everybody responds correctly to it, then you have a situation where it's less likely to spread, but you always bring people in.

Some kids might have them. But it's not going to go someplace else because you've already patched, if you will. But if you have something that is unpatched and it's part of the infrastructure, which is what we got, like the, the SolarWinds Log 4J, then it's on steroids because now someone has broken into something like that.

Well, that tells me is a couple of things. Is that, first of all, do I not trust anybody? if I have to trust them, is there a compensating control? [00:14:00] Do I demand an SBOM, a software bill of materials? I'm kind of getting ahead of our material here. but if we think about it,

[00:14:07] **Cassie Crossley:** a point.

[00:14:08] **G Mark Hardy:** what do we have in terms of frameworks and standards?

Is there anything out there we could point to that somebody

[00:14:13] **Cassie Crossley:** Yeah, so there are definitely lots of, I would say different kind of frameworks. You've got, for example, the NIST, CSCRM

framework, which is, was originally developed for federal agencies, but it's really good for other groups. when you're determining cyber risk, I think that, and I wrote this book, not only for, but it's also good for, you know, for, for, for, for people who are, Enterprise and medium size, but also startups.

So that's probably overkill for a lot of startups to consider. And so that's why I included specific controls in there to pay attention. You know, if you're investigating suppliers or doing this, or you're developing in these environments, this is what's important. You can mature into these other frameworks.

There's COBIT. Frameworks and everything else, but from a CISO, [00:15:00] you really need to understand what you're purchasing or, you know, a CIO or anybody else is like, this is a cloud product. So I have information in there to talk about. I think everybody probably, from our background knows about the SOC 2 type 2 report, but you probably don't realize that it doesn't cover secure development at all.

Right? So, it covers infrastructure security, but how are they building and creating the product that goes with it? It's not in there. So, that's really what's important to this

book And

you know, that's the security and the controls that you need to understand is, you can trust them. They've been audited from an infrastructure standpoint.

But if, if they're just doing vulnerability and pen testing of the outer layer and, and all that, they're not testing, even some of the simple, weaknesses available, such as cross site scripting or, you know, things, there's different kinds of things [00:16:00] that are internal to the system with somebody with the right access controls that they might have fished from somebody could easily bypass a lot of the other controls because secure development wasn't part of it.

and that's, And that's again why I really had to write the book is understanding I work so many people with so many people from IT security and explain to them from a development standpoint, and then I work from people with development backgrounds, but they don't understand threat modeling and, you know, you know, the complications of defense in depth and everything.

And you really have to look at it holistic. And that's very difficult for any size company to say end to end. What do I need to look for and look at for this

supplier? So you mentioned, you know, a couple of times, Microsoft, they've got, you know, a SDL. They were one of the first that published a SDL out there.

Yet, we've still seen attacks on Azure. We've still seen Outlook attacks because there's configuration issues. [00:17:00] There could be operational issues. There could be problems in the DevSecOps, you know, of how they move code through the system. And As a consumer and a purchaser, we're never going to be able to validate that.

So to answer your questions, really, you really have to understand your inventory of what you're using and the critical nature. I'm, I'm surprised every day when, you know, let's see, we're dependent on our cell phones and, and Microsoft teams or other environments. And what happens if those are down for two days?

You know, what happens if they're down for a week? Do we even know how to call the right people to get things going? And some of those basics that I think that some of us who did tabletop exercises, you know, 15 years ago that we knew, you keep a contact list, you do this, you do that. To go to some of those manual processes.

We've gotten out of that. We're so dependent on technology, that, it's very difficult to be [00:18:00] able to bring that back in. And that's one of the things that I, that I talk about is having that business continuity plan and understanding their supply chain, but also making sure. That you have, for example, if you're looking at suppliers, you are asking them these questions.

I want the questions to be asked. I love it when the questions get asked to us because that means the customer requirements are increasing and the posture and the, you know, what's expected is increasing. And that means that the supply chain is also going to improve over time.

[00:18:36] **G Mark Hardy:** and it suggests then that if this is something you understand and incorporate in your own product development, lifecycle or service development, that you're going to get a leg up when organizations and customers start to include these requirements and other vendors are going like, Oh yeah, of course we do that.

We'll prove it as compared to, well, that's been part of our core way of doing things [00:19:00] that we consider as a competitive advantage. Because a lot of times what we find out is that we get a software development life cycle, the

traditional SDL where he says, okay, fine. And then we, we've gone ahead from the good old days, the old waterfall approach back in the, in the government where here's your requirements and then boom, specify them.

And then you go ahead and you go up to the North pole for three months and you write a batch of code, boom, here it is. well you did what we said, but that really wasn't what we meant. Can you dig it? Okay back up to the north pole again. Boom. Here it, well, and then we said, okay, let's, let's tighten that up a little bit.

We'll go to Agile. We'll create little scrum groups and we'll fix that. And now of course we were in like in a DevOps or DevSecOps as we hope to have, where, and again, as I tell people Turn off your automatic updates on your cell phone just for a couple of days and then see how many things get changed in the last 24 hours when you go ahead and you take a look at the app store or something like that.

And for this guy I have 28 updates total are pending [00:20:00] and I just updated everything, with the last two days or something and not a big app guy. It's not like I got a ton of things in there. But what happens then is because our software supply chain is not. Package it up, ship it out. And then like the good old days of mainframe.

Well, once a year you had a big update yet and apply, and that was the weekend you spent all night working things and it was going, it's happening all the time. But now what we're saying is in addition to the software development life cycle, there's a secure development life cycle. When you say a secure development life cycle, is it a secure software development life cycle?

Or does this exist independent of software? And it's really a different way

of just

[00:20:38] **Cassie Crossley:** it's just as we hear the term, DevOps and then DevSecOps, secure software development lifecycle. Really, if you put it all together, that's the way it should be. It should be an integrated part of the DNA. So, well, you know, I come from both worlds, right? the waterfall, rational unified process world.

And then I also come from the agile, [00:21:00] and they both had their advantages. So, yeah, The advantage of Waterfall is, you know, we were very specific about requirements, and how they go through it. And with security

requirements, that's the same thing with an agile world, which is great. You know, you're getting to that minimum viable product very quickly, but you don't necessarily always include the scalability and all the security pieces.

So it's very important that you have. A team that understands not only what security means, but in part of those processes. So, if you're doing a sprint, you know, it's not all about the functionality from a consumer standpoint, but you're also including those activities that are needed from a security standpoint, such as if you're fixing bugs, you're also.

You know, correcting vulnerabilities, scanning, you should be doing scanning, like any kind of, you know, build or, anytime that something's committed, it's scanned before then and evaluated, by these tools to see if there's any known vulnerabilities or [00:22:00] anything in that before it's committed. And otherwise it'll block the build or block the, the deployment or, releasing, anything that you needed to do.

So what's happening today is I'm still seeing. That either it's especially, you know, maybe smaller companies, they're not following an SDLC. They're doing agile. It's very sprint oriented, but they're not taking as much care to do the threat modeling, which is a significant practice that you, you know, to really evaluate the security posture of a product and they're not evaluating the security requirements.

So this could be as simple as, you know, what's, You know, what is the, am I going to use, certain hash algorithms or how am I going to do that? What are we going to do from a technology standpoint? even there's still software that's being released every day that isn't signed. So again, it goes back to that integrity.

It's like, how do you ensure that the [00:23:00] integrity has not been lost during that process? And that was a typical. In the normal software development life cycle, but as part of the secure development life cycle, it's really out there and I really recommend if somebody isn't familiar with and hasn't adopted either NIST, the SSDF, which is the secure software development framework, the version 1.

1, or if we are under the IEC, ISA, IEC 62443, which is an actual standard and the 4-1 you can certify. your software development life cycle too. and those are for if you're doing industrial control systems, but really anybody who really wants that certification and that third party audit. but it has to be part of that DNA for that group.

You can't add it on later. You know, when I, you know, looking at products and you have to do this with open source, that you're scanning it and evaluating it and do code reviews and all of that. And there's no coding rules, secure coding rules. There's a difference [00:24:00] between coding rules. And secure coding roles.

And part of that is understanding, you know, for example, people might have heard of the OWASP, which used to be for web applications, some, some OWASP, yeah,

[00:24:14] **G Mark Hardy:** it's a W.

[00:24:15] **Cassie Crossley:** it's called the world, you know, it's a world application security project. And the reason is, is because they have great top tens for API.

Now, they're doing a draft one for API for AIs. They've got the API, they've got the hardware. I mean, this is information. That should be and we teach it at our company that should be taught to every single software developer and anybody putting together anything is you need to understand the vulnerabilities coming in and that's why it's different just like they need to learn how to understand their languages.

They need to understand the attack paths. And we need to do a much better job on that, and that's that concept of shift left, which I know some people don't like that term, but you're, you're [00:25:00] putting it up front into the cycle, in the design, in the requirements, in the threat modeling, and with the people who can make a difference and not have to include it at the end or when they find a mistake.

[00:25:14] **G Mark Hardy:** And there's some great wisdom in that. Now, what's interesting is we look at all these little parts and pieces that we're going to put together. And as you said, the DNA change is important on the culture because ultimately you could specify a whole lot of requirements internally for your team will do that.

But if they're not enforced and it's not something that there's a little bit of a peer pressure on, and this course, this thing's got to be out the door by five o'clock Friday, it's going to go out the door one way or the other. Now, as we look at those of us who have software development shops, where a lot of organizations do a lot of times, sometimes web oriented, but not otherwise.

The thing I like about your book is because when you're coming out here in 2024, you can include things. We're talking about source code types. And I'm talking about your chapter five, where you have open source, where we just go ahead and grab a library. Okay. OpenSSS. All [00:26:00] commercial where I go ahead and we pay for it proprietary, we write it ourselves, although we'd like to think we do a good job.

It could be Bobby, the intern, who has wrote that code over the summer and it's got some real problems with it. We've gotta rely on the os, the operating systems and the frameworks. and then we got the last two things that you include, which I want to go ahead and just point out a little bit because they may not be as intuitive as the first.

The first one, first one is low code, no code. And the second one, which they said is great because when you're coming out now. Generative AI source code. So let's talk about

low code,

[00:26:28] **Cassie Crossley:** There are code platforms out there, in the, where business teams, non developers have the abilities to be able to create, let's just say either small apps or something like that, or it can be actually, you can have dedicated developers that build it off those platforms. And what that's intended to be is that there is a overall base platform and infrastructure that.

The teams rely on. So you don't have to build the basics from the [00:27:00] beginning. It's very, I, I won't, I may be saying the wrong word for people who understand original object oriented, but when you think about, a business person being able to say, I want this data field and move it over here so I can display it.

So it's really meant to be easier for them to not do any coding at all. and produce some application. Now, what we're seeing, though, is that is very common in HR environments and other kind of finance environments. And a lot of those, business applications and usage are also using PII and certain data.

And. Those non developers don't really understand about access control. They don't understand about data protection. And so it's like, oh, I have this data. And all of a sudden, you know, it starts generating these reports. And things that as, you know, anybody who built business [00:28:00] applications. You know, as part of their past life, know there is so much to worry about in that standpoint.

So I wanted to specifically mention that because there has to be, you have to consider it as an avenue for someone. Let's just say a malicious actor who's entered your environment to make use and go straight to those environments, because no one's evaluated them. They've not done any testing on them. they're just under the radar.

It's, it's not even shadow development. These are approved systems, but there's no gates and checks and balances to make sure that what they're doing is what we would have to do as. As you know, business application developers normally in the past. And that's why I wanted to mention that in the book.

[00:28:52] **G Mark Hardy:** Yeah, good point. Now, and then, of course, the other thing I mentioned about generative AI source code. So I saw something in the press today, I don't know if it was the Wall [00:29:00] Street Journal or whatever, but kind of urging people to move away from using older languages like C because of the memory management problems.

But this is beyond that point. It's just a matter of being a prompt engineer. You are a developer, with all the, you're a developer with 20 years of blockchain experience, or something ridiculous like that. But the point is, is that they, you could get this thing to code, and I've had to write some stuff for me, simple things just to.

Demonstrate concepts and pick a language, JavaScript. I even got to write stuff in APL, which was kind of interesting because I'm surprised that somebody dug that out and they put that in there, but it's part of the ingest. So what's the issue about generative AI source code beyond an interesting idea, which is from an intellectual property perspective, if you want to go ahead and patent, trademark, or copyright something, and it was came out of generative AI, you may have already pre invalidated

[00:29:49] **Cassie Crossley:** Right.

[00:29:49] **G Mark Hardy:** your intellectual property rights.

[00:29:51] **Cassie Crossley:** Yeah. And, and

you said that there is actually a legal suit right now that I reference in the book because generative AI was not [00:30:00] providing adequate, Let's just say not, not acknowledging the open source developers that were part of that. They could detect through signatures that their code was used in what it generated.

So you have legal implications, which, you know, what could that mean? There are certain licenses that, based off the declaration, you have to provide that source code. So if you use that generated AI source code, and it included one of those licenses. Inside of your proprietary code, you might have to disclose the entire source code.

so that is first off a big concern and quite discussed and being followed in the industry. Now, if they were properly. Acknowledging that maybe the lawsuit will, you know, change over time and I know that the different co pilots and, and generative AI, they're working to better identify and attribute the, the [00:31:00] data for where that came from.

But one of the, there's two other points, one of which I did not include in the book. I thought of a better example, later after I released it. But the first in the book that I talk about is that when somebody is generating code. It's learning off of a database that is fallible. It has mistakes. It has vulnerabilities.

So we should not expect. That code to be the same code that, you would expect maybe a 20 year, you know, veteran in the industry to write. And so an example, I gave someone very recently is imagine you're at, you're writing a paper and yet the only training data is everybody between kindergarten and 12th grade, 12th grade.

And so your paper includes. You know, the, the quality out of a kindergartner's writing and a first grade's writing and a second grade's writing. In addition, you [00:32:00] know, it's not just pulling from those senior experts and the, you know, educated and that's the same with generative AI. It has no way to classify.

And figure out which is good software and which is not. Not just vulnerable or malicious software, but quality of software. I mean, expert developers can write, you know, something within, like, lines of just lines of code and somebody else, it would be a hundred. Right? Because over timer, you know how to do it.

And then, you know, so that's one of the things that's very important about generative AI is you cannot assume that it's any better as the intern.

[00:32:45] **G Mark Hardy:** And that is, in my opinion, a profound insight. Let me repeat that back to make sure I heard it correctly. If you're training on data and let's say, for example, we're training on from a kindergarten to a 12th grade and all the papers or writings have been put in, the genre of AI [00:33:00] doesn't discriminate necessarily to say, well, I should.

Do most of this is 12th grade, but it's okay because all that ingested. So it's all going to mix together. You're going to blend. And of course, for those of us in business, sometimes we feel the people we hire are writing that sometimes at the lower end because of whatever system in their education, but that's not part of the call from a software perspective, but you have all these things that you could write the techniques of writing code, the ability to go ahead and put an.

expert who wrote this stuff as compared to, if you will, the intern who came up with this, they're almost equally weighted in terms of, as you look at these values in here, so you're likely to get one or the other, which almost makes you think about the other question that came up with regard to that, is as we find out when we ask it questions, generative AI has a tendency to hallucinate.

Does it do that when you ask it to write code? Do you say, draw me a, write code that draws me a picture of a duck and all of a sudden you end up With the pig.

[00:33:54] **Cassie Crossley:** I haven't seen a lot of cases are very, yeah, so

I haven't that,

[00:34:00] possible and they can, of course, create vulnerabilities for it to be able to produce, but from a hallucination standpoint, I think we would just contribute them more to error and not understanding the context rather than hallucination like we see in some other areas, but that's what I'm going to take away with and watch for now.

[00:34:24] **G Mark Hardy:** now one of the things we had mentioned very early on was the concept of an S bomb or a software bill of materials. And that stuff came out from the White House. And we've heard a lot of it. But what does that actually mean when it comes to supply chain security for our software? And is it In fact, the magic key that unlocks all of our concerns, or is it just sort of a feel good type of an item that gives us a sense that, yeah,

[00:34:51] **Cassie Crossley:** is it okay to answer yes to both?

[00:34:56] **G Mark Hardy:** Yes, you certainly can.

[00:34:57] **Cassie Crossley:** SBOM community, [00:35:00] for now, I guess, coming up on five years. the reason I joined is a customer, especially the utility industry added it to their contract templates. And I'm like, what's in this box? So I immediately joined the NTIA group, and started participating in those

activities and, starting in the, January of 2020, before the executive order came out, I required all of our products.

that we're going through a release to provide the binaries so we could generate SBOMs. Now, I'll go into the quality of SBOMs in a minute, which will answer, you know, some of your questions. but the reason why that's, that's important is, I see folks that, especially, you know, suppliers, I, you know, I, I'm not comfortable giving an SBOM and so on.

I'm developing these and generating them for thousands of products. So if we can do it at scale, other groups can. [00:36:00] The quality, however, has a lot to be desired depending on when it's generated and which part of the process is generated from. So let me explain. So if you are creating a product and you have access to the source code and you have what's called a build pipeline, and you've got a tool that can actually say, Oh, you know, they are using Linux, you know, this version exactly, and so on and so forth.

So it has the data to be able to inspect it. Now, it doesn't give you runtime information, so let's ignore that for a moment. But let's just say an application, install. exe, you would be able to figure out what's in it during that build time of what What software open source packages are there and maybe even the commercial packages.

So we talked about about that commercial libraries earlier. So when I'm doing a binary scan, which means I'm taking that executable install. exe and I'm [00:37:00] running it through a tool and it's trying to inspect it. It's using what's called signature. That looks like it might be Linux, this version, you know, 7.

3, but it could be 7. 4. but I don't even really know, so I'm just going to leave it off altogether. or I'm going to call it 7. 3. So there's errors because it can't figure out the difference because there's such tiny minute changes. And you don't have the hashes to be able to tell the difference at that point.

And secondly, commercial libraries. so if I buy a crypto library from a company, a commercial crypto library, I know that I'm using it, but in these scanning tools, they're meant for open source. They don't have all the commercial libraries. They can identify the Microsoft. net libraries and things like that.

Those are pretty common signatures, but proprietary code that I write. A special reusable components or the proprietary code of a commercial vendor that I've purchased. [00:38:00] It can't acknowledge it. So I see S bombs if it's straight out of the build process. And it's all open source. It's accurate. If it has any

commercial code in either place, you know, whether I'm doing it at that point, it loses the fidelity and it's not as accurate.

And so before we provide a software bill of materials to our customers, I ask the development teams to validate it. to fix the missing holes, to put in the false, you know, correct any false positives, but also false negatives, which means, you know, missing information, so that it's there. Now, can somebody use it for a full diagnosis of a problem?

No, because now the next problem with SBOMs is a lot of them are only being able to tell, here's what's called my first level dependency. So, if I bought that commercial library, And, let's, let's use the microsoft. net. It's a compiled [00:39:00] DLL, right? It's in a software package. I don't know what's in the source code.

They could be calling log4j unless they provide me an SBOM or some, I have some materials that say maybe from a previous security disclosure, what, you know, what potential open source may be in there. I have no idea. So it's a black box to me, too. And that's called a transitive dependency. That means that there are additional fourth party and fifth party dependencies in this SBOM.

So SBOMs are not going to give the final answer to the quiz. They won't tell you if you're impacted, but they will give you some guidance as to, should I start to go look at that? And so I think it's important for everybody to be asking their suppliers for their software bill of materials.

[00:39:50] **G Mark Hardy:** And you mentioned an important point about the transitivity or trust being transitive, and I say it is not, because if I trust you and you trust [00:40:00] Bill, it doesn't necessarily mean I should trust Bill, because trust has to be built up individually, so it doesn't trans, it's not like A is greater than B, B is greater than C, therefore A is greater than C.

We've got good rules in math that allow things to go ahead and be trans, the transitive property that works. So if we have good Something that we figure SBOM helps reduce our risk. It's not perfect, but it's better than nothing, and it's getting the discipline of everybody going. And some of them will be a lot more accurate than others.

And we also have a process where our software development life cycle, we inject security, we're doing threat modeling, and we're doing risk and everything else like that. And then we come up with something and boom, here it is. And

so this is our great product. And of course, we'd like to charge for things when we can, at least.

A lot of corporations do because that's how they stay in business. But then we see counterfeiting, whether it's counterfeiting of a physical device or manufacturer, everything from a USB cable that we look at some of the things that are made to extremely high tolerances and other ones, you look at them and you go.

Wow. I I've seen some nice little [00:41:00] kind of YouTube videos where they put in there and they said, this is just blobs of solder in there and you think it's supposed to have all these wires and it really doesn't and the like. From a software perspective, how do you spot those particular counterfeits that someone might've slapped together?

because the SBOM itself doesn't really represent a reverse engineerable bill of materials where I could say, here's my SBOM and here's my product. I can't go ahead and put the product in a decompiler and have it come back to these values. So I have a questionable product. It's real or it's counterfeit.

I think it should be this. How do I know if I've got it? Is

there, are there any tools that let

[00:41:36] **Cassie Crossley:** the, in the, book about the integrity and how you can validate that. It depends on the different instances. There's ways to do hash. There's ways to check signing, you know, and certificates. The important thing for validating all that, as, as I mentioned, is so that the. The proper secure development life cycle will provide integrity checks, but [00:42:00] also give customers ways to validate that either.

It could be. And let's talk for a minute about hardware. I mean, there's trusted platform modules. There's a different ways besides just software checks that you can have to prevent the integrity. So it can be done at the hardware level. it can be done through software levels. so when somebody is developing products, they need to see, you know, what's the best check.

It's changing every day in the, application security world, especially in infrastructure, security and other areas. There are still, unfortunately, lots of areas, you know, that the integrity checks are not there. How do you know when you hit, you know, a website or salesforce. com or something like that?

How are you validating the integrity of that? And I think that some of us, especially, you know, we have come to realize that you have normal. Kind of bypasses that can happen and [00:43:00] we have defenses for that. So that type of squatting and other areas, I mean, think of that, like a software counterfeit, right?

So if someone's type of squatting or doing areas, that they're trying to, change the route of something like that. But one of the things that I really cover is I want to make sure that. the integrity is not missed during that life cycle. So what happens a lot is developers finish and they've signed the product and so on and so forth.

And let's not, let's ignore the SolarWinds, which actually they signed malicious code into the product. Because, you know, that was, somebody got into their build system. But let's say they've signed the product. How is the, manufacturing site checking that? I, that's a question. I mean, right? Because it's, it might have, that code probably went through a couple different channels.

I mean, remember the days where we would send it in CD over to the manufacturing site? but now it goes through these different pass and distribution [00:44:00] channels. All of them are subject to attack. And we're going to see more of those every day. Where those intermediaries, it's not getting checked. And so that produces a type of counterfeit.

It can be replaced. from a software standpoint, not only do you have to worry about the hardware counterfeits. so let's say you're getting, as you mentioned, a pacemaker. Right? How do you know what's the traceability, what's the provenance, chain of custody of all of that through the process to make sure that the moment that that firmware on that pacemaker was done, not let alone, there's always a mobile app now, right?

Sort of scary. that has got some Bluetooth capability. but, so it goes through this process,

[00:44:47] **G Mark Hardy:** And it was Dick Cheney who said, I do not want that

in my pacemaker

[00:44:51] **Cassie Crossley:** right.

that's happening. Yeah. So, so all along that supply chain, it's going through shippers. It's going through different areas. [00:45:00] It's going to different distribution portals. Like if you're putting up the executable, like if I want to get a firmware upgrade for my printer, you know, I go to their portal and how do I know their portal's not been compromised?

It's a real portal. So there's that risk and that counterfeit and integrity that happens. Yeah.

[00:45:22] **G Mark Hardy:** And that's enough to keep us awake at night. And we could talk a lot more about that, but I'm looking at our clock and we've used up our 45 minutes. And I think we could talk another 45 minutes more of fascinating things like that, but, Cassie, your book, Software Supply Chain Security, hold up one more time.

Available on Amazon. We'll put the link in our, show notes so you can find it. But any last closing thoughts before we wrap up?

[00:45:48] **Cassie Crossley:** Even if you are not a software publisher or manufacturer or anything, I wrote this book so that CISO's procurement people, legal, business, can understand there's a whole [00:46:00] section just on if you are identifying suppliers. You know, this is the things to look at and it provides enough guidance. You don't have to be a technical expert.

So I would encourage you to buy this for everybody in your organization because this will give them the baseline so that they don't have to shy away or be afraid if the word cyber is in something. So I really want this to become democratized where it becomes as common as asking, Okay, you know, what's your financial situation?

It should be, what's your cyber situation?

[00:46:35] **G Mark Hardy:** Wonderful. Well, you can't see, thank you so much. I appreciate your time and all the effort you put into this book for our listeners. Thank you for tuning in to CISO Tradecraft podcast. We hope that this has provided you with some additional information to help you in your CISO or cyber career. This is your host G Mark Hardy.

It's been a privilege to be with you. And until next time. Stay safe out there.