Allan Friedman: One of our participants is a large bank, a globally-recognized bank. They already are asking for a Bill of Materials from each of their vendors, just to see if the vendor can supply it. And if a vendor can't produce a list of what their third party dependencies are, the bank asks for five to ten percent off the asking price, off the top. Not inherently for security, but just for cost of maintenance. I love this story, that it really is a supply chain story. If you don't know what's in your supply chain, then why am I going to trust that you have a quality product?

Mark Miller: This is the DevSecOps Days Podcast. The DevSecOps Days Podcast Series is supported by OWASP, dedicated to enabling organizations to create and maintain software applications that can be trusted. And with support from the Sonatype Nexus Platform, allowing companies to automatically evaluate and track open-source components with known vulnerabilities within the DevSecOps pipeline.

This is Mark Miller, Executive Director of the DevSecOps Days Podcast Series. This is our 138th broadcast. If you're enjoying the series, please go to devsecopsdays.com and subscribe, so you can be up to date as we release new episodes.

Open-source components and their use within the software supply chain has become ubiquitous within the past few years. Current estimates are that 80-90% of new software applications consist of open-source components and frameworks. Section A9 of the OWASP Top 10 places components with known vulnerabilities as one of the most prevalent and abused parts of the software supply chain, placing it at a security weakness level of three, on a scale from one to three. Quoting from the OWASP description in A9, "Component-heavy development patterns can lead to development teams not even understanding which components they use in their applications or APIs, much less keeping them up to date."

In today's episode, I speak with Allan Friedman, Director of Cybersecurity Initiatives at the National Telecommunications and Information Administration. Our talk focused on the creation of a Software Bill of Materials, or an SBOM. As we begin, Allan describes his role in the project and what they hope to accomplish.

Allan Friedman: I'm Allan Friedman. I'm the Director of Cybersecurity Initiatives at the National Telecommunications and Information Administration, or NTIA. We're a tiny part of the US Department of Commerce, and our mission really is about promoting a free, open, and trustworthy internet. Over the past few years, we've engaged in what we call "multistakeholder processes", trying to identify areas where the entire digital ecosystem can come together on things that they care about and make progress. So the government doesn't have a vested interest in the

outcome, we just feel that we'll all be better off if the community can find common ground and consensus.

Allan Friedman: Our initiative at the moment is something around software component transparency. Or how do we find common ground on building and sharing and using what we refer to as a "Software Bill of Materials".

Mark Miller: When you and I had our conversation at RSA Conference, one of the names that came up was Josh Corman. Are you working with Josh, or was he part of this?

Allan Friedman: Josh is part of this and for the listeners that know Josh, know that this is something that he's been talking about for a long time. A Software Bill of Materials is not a new idea. A bill of materials is something that's an integral part of, really, almost any industry. You can't buy an engine without learning about the nuts and bolts that come with it, so you can engage in maintenance, and you know what you're buying. Similarly, in the chemical world, everything comes with a Chemical Data Safety Sheet so that you understand, hey, what is in this and what are the risks. You can't even go to the store without buying a piece of candy without saying, okay, here are all of the components so that you can learn how to adjust that risk yourself.

Josh has been promoting this idea for a number of years and he's found fertile ground in a number of areas. Some of them are regulators. NTIA is not a regulatory agency, in fact, quite the opposite. We think this is something that the community can find shared vision and build out and really deploy themselves. And the way we're going about doing that is not going to software vendors and saying "Hey, please start sharing this." Don't focus exclusively on large commercial software companies that are producing things. Instead, we're going in two directions on the supply chain. We're going downstream and going to their customers, and saying, "Hey, what would be useful for you? What would help you better understand what software you want to buy? And then how do you maintain it, how do you ploy it, how can you maintain vigilance to make sure that the software that you bought that was secure a couple of years ago is still secure?"

We also want to go back up the supply chain and help folks that are developing in the open-source world, that are maybe working on smaller packages which are ultimately going to be used in much bigger products and saying, "Hey, how can you understand, what are the components you're using, so that you can manage it and build more secure products that can then have a lot greater downstream uptick."

Mark Miller: When you guys are working on the legislation part, what kind of buy-in are you getting from the government to enforce a Software Bill of Materials?

Allan Friedman: I want to be clear that we're not actively promoting legislation. In fact, quite the opposite. This was proposed in legislation back in 2013, 2014. There was a bill put forward in Congress that said everything the government buys, everything that DOD buys, should come with a Bill of Materials. And the IT industry nuked this from orbit. They did that for a couple of reasons.

Allan Friedman: One, pretty traditional knee-jerk opposition to regulation. Right? Software industry, very dynamic ... the more requirements we put on it, you know, it comes at a cost. Two, and I think this is perhaps one of the biggest objections; I'm not sure how many of even the biggest software companies, five years ago, could honestly say that they weren't shipping GPL code. Transparency does involve a certain amount of risk if you haven't been looking at your own supply chain yourself. In 2019, that's a much better understood risk. And in fact, there are a lots of organizations and companies and tools out there that exist to help companies understand what they're developing, what they're buying, and what they're shipping. So that issue is not quite as critical as it was five years ago.

The final reason that I think there was a lot of opposition to putting this in regulation is, I think, a very common idea. That we shouldn't regulate without having a clear understanding, without having a clear standard of the issue. If there isn't a commonly accepted way to say, what is a Bill of Materials and how do I use it, then putting it in regulation is going to actually cause a lot more problems that it's otherwise worth. Because it makes compliance either very expensive or useless, because now people are going to have different expectations of what they're required to do.

The approach we have at NTIA is that, let's bring people together, find out what are the core needs, what are the use cases. So that we can start to say, hey, what does this look like on a white board, and what are the existing standards and tools that would allow us to get where we're trying to go.

Mark Miller: What would a Bill of Materials look like?

Allan Friedman: This is perhaps one of the biggest challenges. Everyone who's been thinking about this has an idea in their head, but actually finding a shared vision has been an important process that's taken a little while.

Mark Miller: So what are the common parts between the parties then?

Allan Friedman: One of them is to acknowledge that the core of a Bill of Materials is identity. It's saying, this is this piece of software. Right? This is this library. Now, the challenge is, global namespaces are a known hard problem. There's really only one global namespace in the IT sector that we can count on that works all the time, and that's DNS. For those of you who are familiar with how hard DNS is,

this requires massive global organization and coordination. So we didn't want to generate a solution that required a global namespace.

But there are lots of existing namespaces and so, one, the solution is to say we're going to have every software product should be identified by the tuple, namespace.supplier.product.version. So that's the core, which is hey, it's an identity.

Mark Miller: Normally, everybody can agree on the 80%. What's the 80% that everybody is agreeing on?

Allan Friedman: Sure. The 80% is there, we have it. Which is, this product depends on these libraries. And having a common way of saying, here are the fields that we're going to use, we're going to have the dependency, and then one of the other discussions is ... by the way, what do I, as someone who is making software, how far deep into the tree am I going? And while it would be great to have me going all the way down to all the turtles, each actor is going to be primarily responsible for having the first level down of their dependencies. And then we're going to recurse our way down into the supply chain. That's really going to help drive adoption, even if it doesn't give us the depth at the first level.

Mark Miller: The problem that I would see intuitively from that is a lot of times when we get a vulnerability announcement, it's not the component itself that's vulnerable, it's within the dependency tree. People have to track it down themselves as far as how deep does it go into the dependency tree to get to the real problem.

Allan Friedman: And that's a lot of why we're selling this story as something that has multiple use cases across different parts of the ecosystem. Because a large enough commercial actor, you know a large bank, can basically go to their supplier and say, "Hey, by the way, we want more than one layer". And so now, that supplier, whether an open-source tool, or a vendor, can say okay, let's figure out what our obstacles are.

And so another part where we've agreed on is saying, we're going to explicitly acknowledge when there's an unknown dependency. So for example, I can say my product uses 16 libraries. I've got most of what those libraries' third-party dependencies are, but when I don't have those, I'll just say this is an unknown set of dependencies. So that everyone can see, what are the known unknowns, and that now allows us to do some comparisons. Right? So if I'm looking to buy two products, and I can see that this product has multiple unknowns, and this one, really, there's only a few things that they don't know. And I can crawl down the tree and say okay, yeah, that's a library where I'm not too worried about your lack of visibility of [inaudible 00:11:36].

So we're trying to make visible what we don't know. That's going to be an integral part of this. Especially in the early stages of adoption.

Mark Miller: If people have been following this initiative, one of the things that they'll do is relate it back to Deming's, as far as using fewer suppliers, selected parts, you know the drill on that. Where is you guys' recommendation on this? I know that what you're doing is not prescriptive, but how are you getting across the ideas that are going to be acceptable to everybody?

Allan Friedman: The reference to Deming's is great, because one of the [inaudible 00:12:18] groups that's building out what are the current use cases, and the potential future use cases, has drawn reference to Deming's seminal work on the supply chain in the 60's that really built out Toyota, and then finally the world, to say hey listen, quality is something that we can improve on. And similarly, we're going to see that level in security.

By allowing folks to ask for a Bill of Materials, that itself is a signal. One of our participants is a large bank, a globally-recognized bank. They already are asking for a Bill of Materials from each of their vendors, just to see if the vendor can supply it. And if a vendor can't produce a list of what their third party dependencies are, the bank asks for five to ten percent off the asking price, off the top. Not inherently for security, but just for cost of maintenance. And I love this story, that it really is a supply chain story. If you don't know what's in your supply chain, then why am I going to trust that you have a quality product?

So that's about selecting the suppliers. Once we have that, you can say, okay, let's look at different products from a given supplier. Having visibility in the third-party dependencies allows me to make a certain decision. This is an [inaudible 00:13:35] web platform that we think simply doesn't have the attention, or this is going out of date. So we're going to ask for that to be taken out.

Similarly, supply chain vigilance. I'm using this, now I can actually do some closer monitoring. So for example, in the high assurance case, if there's a low-level library that's used in a very important piece of software that no one's touched for a while, and then all of a sudden, someone's done a bunch of commits from behind a tor exit node, that's something that a very high assurance use case, you're going to want to know about. And rather than making the vendor do that tracking, you can now do that yourself. You can invest in that, or you can work with a contractor that specializes in doing that kind of monitoring. So we're really opening up not just a number of capabilities, but a lot of innovation. Once we have this data, a number of organizations are going to be allowed to build on top of this.

One of the analogies I really like is to think about how CVE changed the security industry. Common Vulnerabilities and Exposures; one, it's a way of numbering things, so that's useful. It's a way of saying, this vulnerability is this vulnerability. But once we have given the power of [inaudible 00:14:47] to the community, now we can hook a lot more products and services onto this. So once we have this data, out and available in the ecosystem, I think we're going to be able to tell a story where more and more organizations are going to be able to say, we can provide value.

Allan Friedman: I'll give you an example. I was initially afraid that the SCA marketplace, the Source Composition Analysis marketplace, was going to dislike this project. Because perhaps by making this something that was provided in the marketplace, we were going to ruin their market. And one of the things that's really impressed me is that most SCA vendors have said not at all. We see this as a market boon. Because their value add isn't just listing what are the third party components. It's the intelligence they're providing about each component. So we're really going to be able to help developers and organizations do a much better job of doing supply selection because that intelligence is already out there in the marketplace. And different sources are going to be provided for different mechanisms and it's going to really help the developers do their job better.

Mark Miller: Last year's major buzzword was blockchain. And it was in the top of the hype cycle all last year. Where does that play into this? It's kind of coming back to reality now, but is there something there with blockchain for security?

Allan Friedman: I am personally a skeptic about most blockchain use cases simply because we have other tools. One of the things that we need to make sure about, when we're talking about a Bill of Materials, is to make sure that we can have confidence in the data. So you're going to need to have some tool. However, we have that already, right? Hashes get us there, or at least get us most of the way. So for example, I'm using this component, and here's a hash for this component so you can validate that in fact, I am using this component.

Similarly, if I'm worried about someone tampering with the data itself, we can use hashes along the way. Blockchain can be built around something that provides third-party assurance, but again, most of the stories we're telling around a Software Bill of Materials isn't public, per se. It's about making sure that users up and down the supply chain can share data with each other. So we don't really need the public nature of it. And we really can rely on some of the existing tools. If someone wants to come along with a blockchain solution, that's great. What I want to make sure is that most of what we can do is easily available with common technology because again, one of the paradoxes of a lot of the blockchain startups is that it does require a certain amount of centralization. And this vision will only work if it's really decentralized. If

everyone is accountable for making sure that their own third party dependencies are shared down the supply chain with their users.

One of our participants is a large bank, a globally-recognized bank. They already are asking for a Bill of Materials from each of their vendors, just to see if the vendor can supply it. And if a vendor can't produce a list of what their third party dependencies are, the bank asks for five to ten percent off the asking price, off the top. Not inherently for security, but just for cost of maintenance. And I love this story, that it really is a supply chain story. If you don't know what's in your supply chain, then why am I going to trust that you have a quality product?

Mark Miller: How mature is the initiative that you're working on?

Allan Friedman: Great question because I think I like the model of Simulated Annealing, where we're not a liquid but we're still a little squishy. There are a number of working groups that have come together to do a number of things.

One, ask what is an SBOM? And you can think of this the white board spec. What does it look like? What's the minimum viable product? Making sure that we have some shared definitions of understanding. That group is working towards publishing their first draft.

Second group is thinking about the why. How is this helping? These are all of the use cases that they're slowly developing. And that is both something that's useful for the future ... people can say oh, this is actually going to be really useful ... but it's also useful for the process itself. To make sure that what's coming out of this initiative is going to actually meet the use cases that we've talked about.

Third, there's the how. So we've got the what, the why, and then there's the how. This is not a standards development process. We're not trying to develop a new spec. But fortunately there are two existing standards in the world today. Both of them come out of the licensing world, because the licensing world has faced a similar challenge, which is how do I say that this piece of software is this piece of software? Because I care about what rules and licenses govern it.

There are two tools that are being used. One comes from the commercial software world and it's called SWID, Software ID Tags. That's an ISO standard. It's really built about making sure that an organization can manage all of the different binaries that come with particular, unique licenses. The advantage there is the metadata sits next to the binary. And so it's pretty easy to track for traditional on-prem enterprise-grade software.

The other spec [inaudible 00:20:00] being talked about comes out of the open-source world. It's developed by the Linux Foundation and it's called the Software Package Data Exchange, or SPDX. It's a much newer standard. Some leaders inside the open-source community have realized that the community itself needs to do a better job of tracking licensing. You've got a licensed product, maybe it's under the Apache license. Someone comes and helpfully commits a bunch of code, but the code they committed was GPL. So guess what, your project is now under GPL. Helping folks who are managing projects track this is going to be really important moving forward. And again, we can use that to also convey data about what are these third components. And so that group is really building out, here's the guide, here's a white paper, and they'll also are building out a fast and quick adoption guide. So if you want to do this for your project, here's how. And ideally, all of these will be based on something like GitHub, so that they're easy to understand and people can write guides for their own products as well.

Allan Friedman: The final initiative is actually a proof of concept. This is coming out of the healthcare sector. The medical device community has been put on notice by their regulator, the FDA, that they're going to be expected to be able to provide a Software Bill of Materials. They wanted to be able to make sure that they could actually prove that this was something that they could do and also use the data. A handful of medical device manufacturers are working with a handful of very large hospitals to generate, share, and integrate the data. And so that is going to be a very powerful lesson learned that they'll be able to feed back into this initiative.

So where are we? Our next meeting is April 11, in Washington D.C. That meeting will be webcast, there will be a Callbridge, so anyone can participate. It will have updates from each of these working groups. My goal is to have, and to sort of help nudge each of these working groups to have, a workable draft by early summer. And that's really when we're going to need a lot of feedback from the broader communities. To say, hey, you forgot about this use case. Or you know what, that won't work for my particular type of software. Or here's something that you haven't thought about. So we're really going to be looking for a lot of feedback at that stage.

Mark Miller: At this point in the maturity cycle, what kind of help do you need from the community? Do you need individual bodies on board, do you need companies on board, what are you looking for?

Allan Friedman: What we're looking for right now is a little bit of coverage in areas where we don't have a clear understanding. We need, essentially, a lot of gut-checking. The Linux Foundation has been a very active partner. Are we covering enough of the open-source community with Linux Foundation, or do we need some other

voices because they have different needs? So that's one area, of making sure that we've got enough use cases from different sectors.

The other area, I think, where we can really use some help and engagement is thinking through how this is going to affect software that ultimately isn't on-prem. What does it mean further back in the development cycle to have and use this data? Ultimately, if the end-user of the software is some sort of cloud-based product or service, what does the customer in that case need to know? And that's a slightly different question. Because it's not just the linear approach of saying here are all of my dependencies, we also have to think about the stack as well. How does this fit in? How can you actually show that this was the code that was used at a certain time? There are a number of folks that are trying to tackle this, but we need some more skull sweat on that particular issue.

Mark Miller: Where do commercial vendors fit in? I mean, I know of tools, and you know of tools, that are helping to do this kind of stuff. How does that fit into your model?

Allan Friedman: The commercial vendors are going to be absolutely key on a number of things. So one, there's the folks that are making and building software. That's really one of the core areas where this is going to be useful. Because the folks who are building commercial software have customers. Those customers are paying a lot of money, and so that's really the supply and demand story we're trying to sell for adoption. The story here is we don't need regulation, because the markets can help. So we're going to be looking for folks who are buying software to say, "Hey, if you could ask for this, would you ask for this?" And so far, we've gotten a very positive answer.

Then we've got the folks that are actually helping other people build software. And that's going to be a huge story. Everything from your build tools ... so GitHub today will actually tell you what your dependency graph is, and it will flag known CVEs if you're trying to pull in a library. That's a great start, but we need to sort of integrate this into more and more of the product. I know that things ... like there's an organization ... someone's built a tool that will build this from Maven, where it will sort of give you a Software Bill of Materials in both SWID and SPDX. That software is available today.

We've talked with folks who are doing a lot of the IDEs to say how can you make this a reality? And similarly moving up the stack into the folks that are providing some intelligence and services for the DevOps world to say, how can you help people actually building this? One of the best examples I've seen is there are a couple of tools out there that actually will say, as you type, the include line will in real time go and flag whether or not we think this is a risky include.

Mark Miller: The final thing, I would say, is how do you want to leave this discussion? Where would people go to learn more about what you're doing and what specifically would you ask them to do?

Allan Friedman: This is a community-driven process. And so if you're interested in what we've been talking about today, we'll have a link to the NTIA website where you can sort of get the documents that have been shared so far, and some information about what's going on. I mentioned the April 11th meeting. If you're interested, it will be webcast. A video will be posted of it after the fact and we'll post some live notes as well, including [inaudible 00:26:02] flagged, and what are the open questions.

Moving forward, really starting to think about if this data was more available, how would this change? What would be better? What are some new ideas? What does innovation look like, once we can build on this? Those are things that will be very helpful for us and the government to know. We want to continue to be able to tell this as a story of innovation without regulation. And so, anyone who looks at this and says I could really help my customers with this, or I am someone who uses software, I want to ask for this tomorrow; having that story in our pocket is going to be really helpful for us as we start to take this, not just around the country, but around the world. After all, this is a global software market and when we work with our government and our industry partners around the world, we want to be able to sell this as something that is really community-driven, where the government is only a facilitator. The community is going to drive adoption.

Allan Friedman: The final area is sort of understanding how will this be different in different sectors. I'll give you an example. I've talked to some folks from the auto industry. The auto industry is in a unique space for software because they're accountable. An auto manufacturer is responsible for everything that's in their car. Whether it was built by the manufacturer or not. Right? They assemble everything, they're accountable. Now for software, that's put them in a bit of a bind. So while the auto industry has traditionally pushed back against required transparency, because it is a very litigious society in the United States, and so they're often the victims of lawsuits, and they don't like to share more information. But they want to be able to know what they're shipping, since they have the ultimate accountability. So that's an area where we're seeing some real interest from a particular sector, for a particular reason.

As we've worked with healthcare and finance, we're getting similarly interesting, unique stories. So for folks who are looking at a sector who has unique demands, whether it's energy or higher ed, I'd love to hear your stories.

Mark Miller: Is there a cost, is there a buy-in, if a company wants to participate as an advisor, or part of the core community?

Allan Friedman: It is free and open. The only thing we will take from you is your time. And I'm aware that that's a big deal. We've tried to make sure that each working group operates as transparently as possible. If someone wants to engage in one of the working groups, they can just join the mailing list, join the phone calls. I'm very happy to jump on the phone and get anyone's team up to speed. I think one of the most effective, in terms of who should be participating, someone who really understands their organization's role in the product. So if you make software, the best participants are the product security people. If you're someone who focuses on things about development, that's someone who really understands how development works and can help nudge people in the right direction ... actually there's a difference between building software and packaging software. There's slight distinctions that we need to make sure we're capturing in this. So it's really about trying to bring as much expertise to bear.

One other model moving forward is to start saying what are some of the upstream and downstream focuses where they're going to be using it? So I've been working hard to reach out to the vulnerability management tool providers to say this could potentially really help your business because now you don't have to do the registry mining to figure out what's on a server, what's on an endpoint, because the vendor's going to tell you what's in it so you can actually do a better job managing things.

Allan Friedman: So trying to understand what are the choke points to adoption are going to be key. So that's the downstream side, is how do we help enterprise customers use this data? Moving upstream, I don't have a clear understanding of where's the open-source supply chain opaque? Where is it not [inaudible 00:29:58] when I run a make with a couple of flags, I'm pulling in everything that I'm using in my build process. So it should be pretty simple.

There are going to be certain parts of the development world where that's not easy. Where we're essentially switching languages, we're switching platforms, and so there are going to be certain obstacles to gathering the data. Anyone who can have insight and can say listen, you should start to look here because this is an area where we simply don't have visibility. This is complicated because the open-source community is an incredibly diverse community. A lot of it is driven by the corporate world, but a lot of it isn't. And a lot of corners of the open-source development world really don't want to sit down and talk to someone from the US government. Even if he does have funny hair and goes to hacker conferences. And so having that outreach is going to be another key aspect.

Mark Miller: The idea of getting vendors to, in theory, donate their tools into the swamp, so that government people have access to them to see if they were applicable for their projects. Is that a Phase Three for you? Are you considering it? What's going on there?

Allan Friedman: This is where being a tiny government agency gets a little tricky. Because obviously we want to see this succeed and be sustainable. So once we get our initial guidance documents out there ... again saying this is what a Bill of Materials is, this is why you should use it, and this is how you can implement it in your own organization ... the next phase is going to really be promoting awareness and adoption. How do we push this out into the broader world? And once we're there, we're going to be looking to say what is long-term sustainability look like? Is this something where different sectors are going to take ownership of it from their own perspective? Is this the sort of thing that really makes sense to hand over to a global standards organization, or a global security organization, to say why don't you maintain a best practices list?

Because I think there is real benefit to having a centralized repository of all of the knowledge and all of the tooling that's being built out. We want to start that. Ultimately, my tiny little corner of the Department of Commerce probably isn't the right place to keep that. And so we can work with our government colleagues and our private sector colleagues to say what does that look like so that it's vendor neutral, it's platform neutral, everyone trusts it, but it's still a stable model.

Mark Miller: This is the DevSecOps Days Podcast. The DevSecOps Days Podcast Series is supported by OWASP, dedicated to enabling organizations to create and maintain software applications that can be trusted. And with support from the Sonatype Nexus Platform, allowing companies to automatically evaluate and track open-source components with known vulnerabilities within the DevSecOps pipeline.

This is Mark Miller, Executive Director of the DevSecOps Days Podcast Series. If you're enjoying the series, please go to devsecopsdays.com and subscribe, so you can be up to date as we release new episodes.