TRANSCRIPT FILE

JULY 19 NTIA - PART 1

I would like to welcome everyone for being here today. For those of you who came into Washington DC, thank you for making the voyage. We feel that July in Washington is really the best time of year to be here, so thank you for doing it. For those of you who are joining the webcast, thank you for taking so much of your day to be with us. We are going to work very hard to make sure that you are part of the conversation along with those of us in the room. My name is Allan Friedman. I am the Director of Cybersecurity Initiatives at NTIA, The Department of Commerce. I am going to be standing in front of you for most of today, but hopefully I am going to be doing very little talking because the goal of this meeting is to make sure that we find a way to capture the perspectives that are in this room and the incredible expertise that is in this room and figure out how to make some progress today. A moment on logistics and the plan of today. We are going to start this morning with some shared perspectives from some folks who have been thinking about this issue for a while. The goal here is to put the ideas on the table. This isn't going to be the complete sum of perspectives, but we want to be able to start the discussion. Then we are going to move into the meat of the multi-stakeholder process which is listening and sharing and bringing ideas to bring to the table. We are going to have a facilitated discussion in the morning. We are going to try to talk about news cases. In the afternoon, we are going to flag and think about some of the challenges that we face, and then at the end of the day we want to bring it together and say, what's the path forward? Some of you have participated on multi-stakeholder processes before. We are very impressed that you returned. [LAUGHTER] For those of you who are new, this can get a little messy at times. Bringing this many people together with this many different opinions is going to feel a little chaotic, but it's that value of bringing those shared perspectives that really makes this approach go forward. For those of you in the room, if you are not aware from the cameras and lights and the microphones, this is being webcast and it is be recorded. We have had some requests for folks to post this video once it is done. After today, if there are those of you who feel very strongly that we should not post this, we will raise that issue, but otherwise our

default will be to have that video public so that folks who were not able to make this first meeting can still catch up and learn from what we said today. So please [INAUDIBLE] that. I also want to make sure that everyone in this room knows that this is seen as a public meeting, and there is press in the room, and so we just want to behave accordingly. I'm going to now, my pleasure, to introduce our, not quite new, he's been on the job for quite some time and done amazing work so far, our assistant secretary David Reynold who is going to offer some perspectives on what we're doing here today. Thank you.

Thanks Allan. Thank you to for those of you who are in the room and for those of you on the phone for joining us today. The turnout and responses we have achieved so far are [INAUDIBLE] Sorry to the people on the webcast. Thank you again to those of you that are on the webcast are now being able to hear me. The turnout and responses we have received have been a very encouraging sign. There is real potential for progress on this issue. As most of you know NTIA is the executive branch agency that is principally charged with advising the president on communications and information policy issues and our main areas of focus include domestic and international Internet policy, expanding broadband access in the use of spectrum, and cutting-edge communications research. NTIA also has promoted the use of the multi-stakeholder model to help address a range of policy issues. Today's multi-stakeholder meeting begins a conversation to discuss software component transparency. This is the third cybersecurity related multi-stakeholder engagement that we have convened. Our prior initiatives dealt with coordinated vulnerability disclosure and the security update ability of IOT devices. Those issues and the topic we will tackle today have a few common themes. First, they acknowledge that our digital systems will never be perfect. There are countless efforts across the government and the private sector to improve security, but for the foreseeable future, some vulnerabilities will continue to exist. NTIA's work has focused on increasing our resilience in the face of a constantly evolving risk environment.

Second, we have tried to be timely and take on issues where building quick expert level consensus on a rapidly emerging risk can make a significant difference. Our multi-stakeholder processes are designed to be agile and enable a community to more

quickly find common ground. Compared with typical regulatory or legislative solutions, the multi-stakeholder model can more nimbly address emerging technological issues. Third, we know that not every participant will agree with one another but avoiding contentious issues will not lead to progress. The power of the multi-stakeholder proceeding is the expressing of different perspectives which ultimately helps identify areas of overlapping interests, and that's where the sweet spot of consensus begins to emerge. These processes can appear restless or, let's be honest, even fractured, but as we listen to one another which I know everyone in this room and on the webcast will do, we will begin to understand one another. We will find the common good that rises above the various proprietary interests, and we will begin to make progress. Our room here at AIA is particularly well-suited to this conversation because it encourages you all to talk to each other, and for those participating remotely, we are going to work hard to make sure that your voices can be heard. We are very lucky to have a number of sectors represented here today, software vendors, telecom providers, healthcare, finance, auto, medical device manufacturers. I asked experts in civil society, this diversity can add complexity, but in this age of connectivity, it's only through this type of collaboration and cross sector partnership that effective, harmonized solutions will emerge. This is the real power in understanding the common challenges that we face. The idea we are tackling today, software component transparency, is not new. It's one that has taken a number of different forms in different industries. Tracking with third-party code is used in software products is a well understood practice, although, one that not every vendor follows. Many of you are here to talk about the potential benefits of sharing this data or want to explore how to ask for it. Some of you want to highlight the potential costs and complexities of transparency. Others are already on their way to doing it and are interested in exploring what standards, formats, and practices we can use to further our interests. Today we would like to hear your perspectives on the potential and the challenges of software component transparency. Let's begin the conversation with the potential. What problem can we solve and how can this improve security. From there we can pivot to the potential pitfalls and identify the real challenges in generating this data, securely sharing it and effectively using it. As I've said, these processes are designed to be flexible. You will do the molding, will define

the scope of this process and what meets your needs. We here at NTIA are here to help guide the process. Hopefully over the upcoming months we can help you achieve some consensus around some aspects of software component transparency. You should also think of the outcome of this process not as an end, but as a beginning. As other organizations build on your work and integrate it into other efforts. The goal here, ultimately, is to catalyze change. For today's meeting, we are looking to you to identify goals and create internal structures that will make this process more manageable such as drafting committees or working groups. The goals in organizations can evolve over time, but this will be a useful starting point for this effort. The group also need to identify the location and frequency of future meetings. We have a potential schedule we can share, but ultimately this will be up to you, the stakeholders. Our ultimate objective is to drive the creation of industry-led, market-based cybersecurity solutions to foster a trustworthy and resilient ecosystem. That is something we should all agree on, or certainly hope we can all agree on, as a starting point.

Identifying what to do and finding consensus on how to do it, that's the hard part. We feel these conversations function best when the right experts are in the room willing to work towards solutions that reflect the needs of the broader community, so I think we are already on the right track. I know there is a lot to talk about, so I'm going to stop there and let you all get to the hard work you have all come here and agreed to do. I thank you for your participation, and I look forward to the readout of what happens today.

[APPLAUSE]

Thank you, David. I think now we are going to actually, as David said, let's roll up our sleeves and get to work. We are going to start, as I mentioned, with some of the perspective sharing, and it is my pleasure to introduce Art Manion who has done an amazing amount of work in this space, and these are your slides. I am going let you drive them. From CERT/CC and for those of you who are in the vulnerability space, you have probably worked with Art who has been doing this for 20 years?

Seventeen, close enough.

For those of you, I am going to be working very hard to make sure that we can move the discussions, so I would ask each of our speakers this morning to keep your marks to

eight minutes.

Can I get a browse on the screen? Do you know how to do that? Thanks. Thanks Allan and welcome everyone. Allan already mentioned this, my name is Art Manion. I am the Vulnerability Analysis Technical Manager/Principal Engineer at the CERT Coordination Center. I say with the slash because I am about 70/30 these days and that's just how things are.

To get things started today, there are a couple of areas that I'm going to talk about in terms of software supply chain transparency. Some defender use cases -- at CERT we do a lot of coordinated vulnerability disclosure, and probably the number one question is, "Am I affected? What is the list of affected vendors? Especially for multi-party library and protocol-based issues. I need to know if I'm affected and how to defend appropriately. We have given some thought to this over the past probably four or five years at CERT. We are nowhere near any kind of solution, but some of that thinking has led us to trying to generalize some pieces of the puzzle we think would be necessary, so I will talk about those briefly as well.

This defender use case -- there are a couple classes of defender that we were concerned with. There is the end user, the administrator, the consumer, someone using software, someone relying on software, and again, the main question being, Are the things I have vulnerable? Are the things I depend on, my cloud service, my Gmail, are they vulnerable or not? When a vulnerability is disclosed, there may be actions I should take, patching, some other configuration, mitigation option, or I may ask my provider if they have taken actions. I want to know if I'm vulnerable and if I have done the right things to protect myself. That's one thing when there is a disclosure and there is no immediate sign of attack or exploit activity. Under the pressure of an active attack or something in the wild, that question becomes a much higher priority, and people scramble and scramble to answer. Do you have inventory? Do you know which pieces are tagged with the vulnerable code? There is a procurement and maintenance aspect to this. If I'm going to acquire or procure a purchase software or service, are the things I am buying or procuring vulnerable? It's not a great idea to just count vulnerabilities and say product A has 27 and product B has 5; therefore, product B is somehow more secure. That's not a good count so don't do that please. However, the fact that a

vendor provider is giving you the bill of materials, the mere existence of the SBoM indicates some level of maturity. You have some sense of the upstream components. There may be a lot or a few and very simply, complexity means more vulnerabilities. Even if there are 27 known vulnerabilities, you may know the status of them. Some are patched; some are not patched; some are accessible via your software; some are not accessible. You may choose to accept the risk of those that remain unpatched and that's okay if you are making an informed risk decision.

Another element is sort of end-of-life, the components end-of-life, am I going to keep running it or the vendor went out of business. The last available SBoM might be my list of what I am running, and there will be no further update but at least I've got that information in hand, particularly for long-lived legacy systems with a software and the system outlive the vendor in this case.

There is a vendor use case as well. Same questions essentially. Are the things that I put into the things I make vulnerable? A vendor presumably wants security for their customers, same things. During an attack, much, much higher priority question, much more pressure. Also, procurement and maintenance. I am a vendor, I am taking some things from upstream, I am assembling them, I am adding my own code, I am increasing a product or service going downstream. So, I am still doing a procurement thing. Maintenance, signal of security quality of my upstream providers. There is a little bit more of a supply chain, hygiene element for a vendor. If I have fewer components, better management of upstream components, I have fewer things that could go wrong. Basically, it's just again a complexity argument. Less chance of a high priority break fix when a vulnerability comes out in an upstream component.

There is a third defender use case that I will refer to as critical infrastructure, public safety. This would be DHS in kick [PHONETIC] in my world, in the US and policy as well. "Hey, this new vulnerability came out and it affects lots of things. Is the energy sector affected? Is the electrical generation sector affected? Is water and sewer affected? Is healthcare affected?" We hope that a well-functioning supply chain system could inform those kinds of questions and not just am I personally affected, but what are the sectors involved here? How prevalent is something? Is the formability exposed to attack or not? We are going to get to this but having a vulnerable component does not

necessarily mean the vulnerability is exploitable.

So, three defender use cases. Some of the puzzle pieces. We think there are at least three main components. We would like to see a standard inventory format, so if nothing else, we all provide our list of ingredients in the same format. As soon as there's two formats, we are in huge trouble, and there can't be a thousand things in a thousand formats because that's just not going to work. You folks may be familiar with some of these SWID and SPDX exist. They are in use; they are real things. CPE is also in use and a real thing. I am going to go so far as to say probably not CPE for this use case. I say that in actual consultation with NIST [PHONETIC] folks and giving it some real consideration. I'm not just dissing CPE on a slide here for no reason. Standard inventory format, relationships between components. This is something that CERT focused on before. We found out that the relationship graph is actually an interesting and not too hard problem; the problem is getting the data to fill it out, thus, the need for the inventory format. Component A includes component B. And you need to tie vulnerabilities to these things as well. "Hey. Component B we are sure is vulnerable to a certain vulnerability. This CVE is an old JBoss vulnerability it turns out; therefore, component A is vulnerable because it includes component B." The "therefore" is a little bit sketchy. It could be; it's probably worth investigating. It may be vulnerable. That's because -- just the inclusion of the component does not necessarily mean there is an exploitable path, but that's a hairy question we have to explore a little bit, I think.

Having a vulnerability mapping requires vulnerability identification. Vulnerability identification today is largely -- CVE is the public US sort of flavored version of that. There are other mechanisms to do vulnerability identification as well. CERT/CC, we publish a vulnerability. This is just a screenshot of the bare minimum top 10 or 11 entries for the crack [PHONETIC] vulnerability. I think we had 120 vendors listed. There is widespread belief that our list is authoritative. It's a little bit authoritative. We do try very hard. We have experience in this; nonetheless, this is created with significant manual effort every time and it's incomplete. After every one of these we find vendors we didn't know about, and there are more people that make these lists. I've seen things come up on GitHub or Google Docs, people making their own lists, sort of community network driven lists. Ours is not the only one.

A quick sample of what I consider component relationships, this is probably a little bit difficult to read if you are too far away. There is an embedded Web server called GoAhead Web Server. It's in lots and lots of things. It is in programmable logic controllers; it is in building security little device boxes. I forget what this Radware app detector product is, but I think the administrative web server is also GoAhead Web Server. GoAhead Web Server has a vulnerability. It's fixed by an upgrade to GoAhead Web Server. What's the graph of all the systems affected? This requires, again, inventory, vulnerability mapping, and their relationships. I will leave this example on the screen and end my talk. This is the JBoss CD I used as an example earlier, known and patched in 2010, exploited sadly by Ransomware in 2016. Probably someone didn't know they had it, didn't know they put it in their systems. That's a massive failure of very core basic fundamental security. Didn't know I had it, didn't patch for six years, got Ransomwared. Complete failure. That's our hope for what a software transparency system might be able to solve. Thank you.

[APPLAUSE]

Thanks, Art. It's now my pleasure to invite Bruce Lowenthal who is the Senior Director for the Oracle Security Alerts Group. Let's see if we can find your slides here.

Thanks, Allan. I am Bruce Lowenthal, and I lead the group at Oracle that's really responsible for product security, will be the target, let's say, of many of the things that may come out of this meeting. We will be responsible for ensuring that a lot of the products get this included. We also make policies at Oracle for security, so we are familiar with that, and in this discussion, I am going to talk about some of the things I think that we do when we create policies to make sure that the policies don't have unintended consequences which is often the case here.

The purpose of this presentation is to help ensure that the transparency proposals are evaluated for both positive and negative issues. We don't want people to say, Transparency is good. We'll get it as much as possible and ignore how it might be abused or have outcomes that are undesirable. All defining goals for different constituencies, because different views will have different about transparency and why they want it or not, and we want to look at some of the negative effects, and I'm going to bring up negative effects because I think most people here are going to be talking about

the benefits, and I wanted to have a contrasting view; although I am not against transparency at all. I just think transparency is a good idea, but I don't think it's good if the side effects are improper.

When we do proposals for policies at Oracle, we try to make sure that we define the goals first. What are the goals? What are we trying to do? Why is this good idea? We look at the different groups that might be affected by such policies. In the case of this, we look at least vendors, customers, and third-party component developers because one of the issues that I am concerned with is that if we put too much burden on things like SWID and things like that make it tough for third-party component developers, that we might end up having fewer third-party components to select from in which case reuse goes down, and that will significantly reduce the security and bug freeness of our products.

Here are some examples of proposed goals. I am not promoting these or not. I'm just giving examples so people get an idea of what I mean, and I think it's up to the different groups to define goals. One is to improve production product security while maintaining or reducing disruptions. I am saying "production security" because I want to make sure that we care, Oracle cares, about security in production, not security in other places.

A low level, not low priority, goal might be to produce an inventory of fixes for third-party components that are only exploitable within the context of vendor products and the customer production deployment. So, it would be ideal if we could have some kind of magic tool that would tell the customer, "These are the issues that are unresolved that are actually exploitable in your environment." And right now, we've got a major problem with that because what happens is, people see lists of vulnerabilities and they just assume that they are vulnerable because no one can convincingly tell them that they are not.

Another issue at a high level would be don't provide gratuitous information to avoid unintended, bad consequences. We have seen some of this already, and I will talk about that a little bit later. One of them is, somebody sees your inventory of components and they make a sales call because they want to replace your third-party component with another one. If that's the goal here, then we should state that or not, and you may feel that's inappropriate or you may feel it's fine, but we will decide that

later.
Some people may boycott certain products because they include software from places they don't like or things like that. You can decide whether that's good or not. Sometimes we have seen sales like [INAUDIBLE] problem, which many of you may know, was used as an issue to stop competitive bids from different vendors, and that would be -- those issues would be apparent from inventory lists of third-party components that were in products. I'm not saying anything against that, I'm just saying let's talk about, let's think about, what are the negatives that might occur? I did an inventory on one product at Oracle. It's not one of our largest products. It's a large product, probably something like 100 million lines of code. This was an acquisition a few years ago. It's not like it was developed with Oracle processes and policies that we would do today, but most of our products are acquisitions and that's true of a lot of large vendors. There were 300 third-party components that we saw. The numbers actually that we came up with -- this is larger because it's difficult to understand when you have the same third-party component because there's no naming standard for these things -- I'm just going to say 300 plus, although, I think the number I actually found was 351. Three minutes. Okay. There was 150 ""vendors", and that's in quotes because a lot of these people, or a lot of these organizations I'll just call them, are not what people would call vendors. 25% of the vendors are just named individuals. That's not necessarily bad, but I am just giving you an idea of what you might see because that might be surprising to some people here. Some very important products that are used today by lots and lots of vendors, FPP is an example, were done by single people at one time. Considering possible goals and outcome, vendor goals might be to include software for re-creation and to also satisfy customer goals. Customer goals might be to include faster production inclusion of exploited vulnerability fixes. These are for security or compliance reasons. What we are seeing more and more is customers are just being told, "You have to put this fix in. We don't want to hear about whether you are actually exploitable or not. We have some kind of compliance issue with the industry or some other group that says it has to go in and that's that. This is highly disruptive. Third-party goals might be to prevent onerous regulations that inhibit the component use as I talked about earlier. We also need to consider auxiliary requirements. We need to be able to

uniquely identify products and components or any tables that we come up with will be degraded in their usefulness. This is a huge, huge problem in the databases at Oracle. We see something like 10 different ways to name the same actual component, particularly when you take into account the vendor name and the component name.
Finally, as promised, I will talk about some of the negative effects of transparency, and these are mostly from what we have seen today. Customers are currently using tools to construct third-party inventories, and as a result of those tools, they have made patches per the NVD database or other databases. At Oracle we look at the NVD database every day to see if there are new vulnerabilities coming out that affect our products. Not everyone can do that, and so assistance there might be helpful for many products. Are these tools good enough now? And if they are really good, do we really need to have software transparency because we can use a tool. I would say, at Oracle we found that tools are much, much more effective at finding that than trying to get different development groups to try to figure out what they put in there and keep track of them and put lists in there, not only because they make mistakes, but because they don't know that a component is within another component or just because we can't match the names or things like. The mandating patching has been a problem. A really good example is a log per day fix that came out, I think, a couple months ago. That affected hundreds and hundreds of Oracle products. As far as I could tell, no more than five are actually exploited, and the result is thousands of Oracle customers are applying patches they don't need, and this doesn't help security at all. All it's doing is disrupting their work, and probably in some cases there is going to be mistakes made in which case the systems will not work properly. So, lots of problems for no benefit. I'm not saying we should abandon transparency for these reasons, but we need to make sure that whatever we come up with is something that addresses these issues to make sure that they don't occur or at least not as frequently. Heartbleed was another one. I think most people have heard of that. Same thing there. Huge percentage of company products vulnerable but only 20 of the hundreds were actually exploited.
The last point here has to do with if we cause people to patch too frequently because they see daily fixes going out, what we will find out is that those groups will stop applying fixes and will pick and choose, and they never do a good job at pick and

choosing.
>> ALLAN FRIEDMAN: Thank you very much, Bruce.
[APPLAUSE] I think we are going to invite another vendor perspective, Jim Jacobson, who is the Chief Product Security Officer for Siemens Healthineers.
My goal here is to give you some of the thought, some of the perspectives that a device manufacture has in this space. First, let's start off with some assumptions. Assumptions are assumptions. These are not assumptions that have been proven, but assumptions that we should challenge. One assumption is that manufacturers have all the information that's needed to share in a software bill of materials, for instance. That is gathered from the fact that it has to be available in some format in order to produce the product. It also has to be available in order to do software vulnerability management in general or to monitor for vulnerabilities. Another assumption, manufacturers want to provide information to their customers and transparency in this space goes beyond what we would be talking about for a software bill of materials. There's a lot of other ways we can convey information about the vulnerabilities in the or the software stance of a product, security white papers, industry-standard forms in the medical device space. There is one called the MDS squared that we use. That assumption is there. Why do we make that assumption? It's required in order to collaborate for managing risk and for reducing risk between the manufacturer and the customers of those products. Also, of course, it's the right thing to do to share that information because it establishes trust between the two parties. Another assumption, manufacturers understand some of the use cases for software component transparency. This is an assumption, and it could be that what we are trying to address is the customer's risk management processes, whether that's proactively managing risk, or in the case of incidents and vulnerabilities, a reactive stance that they need to take.
A final assumption that I want to point out, of course there are many more that we can delve into overtime, manufacturers understand some of the requirements and some of the challenges needed to implement some standardized approaches. Things like machine readability, things like unique software identification.
Let's talk about the solution space for a minute, and this is solution space that has come about from conversations with many stakeholders. It's not a definition of a solution, but

it's defining some of the problems that we hope to solve. One dimension that we look at is depth. In a software bill of materials, how many components or how many items are listed? That's the depth dimension. The breadth dimension is how much detail is provided for every component that would be identified and how much that information -- the level of information that each component has associated with it. And the final dimension is time. The time dimension is how often do we update software bill of materials in a way that can be used by our customers. So, in depth we -- the first level of depth that we could look at is simply identifying all of the components that are relevant to describing the attack surface that our customers would be interested in. That's the first step, and in order to -- if we were to leave it at that level, it would be incumbent upon the establishment of trust between customers and the vendors because you are not specifying everything. And then why are you not specifying certain things? It's because there is a trust that we identify them correctly. It could be all software components wherever you define "all." Everything that we would include in our vulnerability monitoring, for instance, would be including the software bill of materials. It could also include firmware and hardware. We get into more challenging areas if we talk about snippets, code snippets that are harder to identify reliably, and then the issue of recursion. How far back in the supply chain are we really going in order to describe what would be in an SBOM. When talking about breadth, we could start off with an identification of the component, of course, the major revision, the minor revision, and then again, we get into areas which are more difficult to characterize, patch level because of timing. It might be challenging to a manufacturer. Other attributes, contextual information, how the component is used. What is the relationship between the component and other components that are present? And then time, the time dimension. An SBOM could be provided at the time of delivery of the product. it Could be provided every time there is a major update to the software. It could be anytime when there's an update of any kind or even whenever the system is patched, we could theoretically update an SBOM; although that imposes some difficulties, just thought [PHONETIC] and logistical approaches. Finally, in real time, that would be the ideal case where this information would be available in real time although there are certainly challenges associated with that. We've discussed all of these options in talking with our

customer base and our goal, I hope as a group here and as this effort continues, is to nail down some of this solution space in these three dimensions. Finally, let me start off with nailing down. What do we want to nail down? By the way, the nail is crooked because things never go as straightforward as you expect them to. [LAUGHTER] We also have to be careful that we are not acting as hammers and seeing everything as nails. But what can we nail down? What should we look to nail down? We should look to the use cases, and we are going to be talking about some of that later on today. We should also nail down unique software identification, availability of frameworks and standards and which ones can be adapted or used for this effort. And then that 3D Solution Space that I was talking about. Thank you.

[APPLAUSE]

Thank you. It's now my pleasure to invite Chris Wysopal, who is the CTO for CA Veracode, who knows a few things about vulnerabilities.

For those that don't know, Veracode is a software screen testing company. We provide technology for vendors and developers to test their software. We have been around for about 12 years. In the last four years, basically, post-Heartbleed, we have been doing open source component analysis. What I'd like to do today is bring some data that we've gathered from our analysis of the vulnerability space and from the components space. I'm going to talk about vulnerability information sources and vulnerability likelihood. It probably doesn't come as a surprise that not all vulnerabilities are in the NVD. There is a lot of public information out there because, of course, we are talking about open source, so there is public information out there in security bulletins, release notes, commit comments, and comments in the code itself that describes vulnerability information. Just because that information is available doesn't mean that anyone has communicated that information to CVE or the NVD. Of course, this information is available to both attackers and defenders that care to look for that information so it's highly relevant. What we do in order to help our customers understand the risk and the components they are using is we crawl all of these open source repo's [PHONETIC] on a nightly basis. If a customer starts using a component and is using our technology, we will make sure that we are scanning that repo on a nightly basis going forward to make sure that we have the most timely vulnerability information, and in many cases, the

information is a security fix. The security fix is put there, now there is knowledge that that release that was fixed has a flaw in it, and sometimes it will actually make it into CVE at a later time, months later, or even years, later, but the information is public. That's how we are able to get it. I went through all the different components that we're scanning. We are scanning the open source repo's for all these different languages. You can see that across -- we are not scanning the whole world, we are just scanning the ones that our customers are using -- but out of all the components where we scan we find 3553 CVE entries relevant to those components and 253 are actually reserved CVEs across there and we find 1689, we call them SVEs for source clear vulnerability enumeration. That ends up in a few -- we assume that some of those reserve CVs may be overlapping with the SVEs. That's why I have a % SVE Low and a % SVE High, but essentially about 30% -- we find about 30% more vulnerabilities in that are in NVD. That I just want to bring to light that the source of vulnerability information isn't necessarily we can't just rely on NVD. Customers, stakeholders, vendors can't just rely on NVD. There is a significant amount outside of that world. The next thing I want to talk about was exploitability. I think we all know that just because you have a component included into your product or application, it doesn't necessarily mean that that makes that product or application vulnerable. The main way it makes it vulnerable is if the product calls into the component and there is a control flow that we can see that eventually exercises the code that has the vulnerability thereby allowing an attack surface that the attacker can exercise to then exercise the vulnerability. We do control flow analysis across three different languages, Ruby, Java, and Python, and we are working on supporting all those languages over time. Right now, over those three languages, we do that, but I also want to point out that sometimes the component itself might expose an attack surface directly and that can happen. So, there is also the chance that just merely including the component does make the product or application vulnerable, although, that's rare, but I don't have any data around that. I don't think anyone's done that analysis. If we look at across Ruby, Java, and Python, this is how many different applications we have analyzed. 510 Ruby applications, 5200+ Java, 585 Python. When we do this control flow analysis, these are the percentage of applications that are actually vulnerable that contain a vulnerable component. From our data, across

these three languages anyway, it's less than 5% of the time does including a vulnerable component make the application or product vulnerable. The only other published data I have seen is from Contrast Security and their product does this kind of analysis at run time, so we do it statically, they do it at run time, and they have publicly stated that they see it less than 10% of the time that inclusion of a vulnerable component makes the application vulnerable. It may because they have a slightly different language coverage than we do, and we'll get better over time as we build up our data set. Just in conclusion, NVD can't really be considered authoritative because it doesn't include 30% or potentially more, at least 30% is what we found, of the vulnerabilities that are out there. The assumption should be that if something contains a vulnerable component, the assumption should be it's not likely that that makes it vulnerable, as opposed to the other way around, which is, I think the way customers think about it is, if you include the component, it's likely you are vulnerable unless you did something special. We have got to flip that around, and the mindset should be, it's not likely to be vulnerable just because it includes the component. I'll end it there. Thank you.

[APPLAUSE]

Thanks Chris. Now I would love to invite up Josh Corman who is the cofounder of security group I am the Calvary. He is the CSO of PTC.

I have a lot to cover so I will go fast. In general, I agree with most of what Chris just said there, but I want people to put this in the context of safety critical environments. Even 5% exploitability when the consequences of failure could be [INAUDIBLE] physical cyber physical harm is a pretty large number, and if any of those are preventable harm, that's one of the reasons there is such a lean forward attitude from DHS for critical infrastructure, from FDA from the international community, so I am going to focus a bit more on the public safety of the human life. I am the Chief Security Officer of PTC. I have a lot of software products. I have to produce software bill of materials for a lot of medical device, oil and gas, critical infrastructure customers I have. I have to eat my own dog food here. To the point, I'm going to focus on some of the public safety human life where bits and bites meet flesh and blood from the Calvary, and more specifically, I was part of the congressional task force for healthcare cybersecurity, and one of our top recommendations which is being acted upon both by House Energy and Commerce

and by HHS is they are actively rolling out a project to do a software bill of materials for all medical technologies. That is happening irrespective of this effort. My hope is that the collective brain [INAUDIBLE] here can identify risks or issues or challenges or scoping levels that could make that better and more consistent across sectors. This is what we are really talking about. Some of the objections you hear tend to be about secondary and tertiary potential uses of this, but we are talking about ingredients list. It's on all the food we eat; it's in every car you buy. This is a common practice for manufacturing. Ingredients is an inventory of the parts. We are not even necessarily talking about nutrition labels, although, that could be a branch in a sequel for what's the attack surface, what's the ports being used, those kinds of things could also be valuable. Is it patchable? Does this vendor have a [INAUDIBLE] disclosure program? There is a lot of FUD, I'm just going to call it FUD, but a lot of the top common objections, I think many of them will surface today, say it can't be done, but it's being done, or that we could never do it, we don't know what's in it, but they are often presenting those because there is legal license requirements for a lot of these open sources to declare what you are using. This is a sample from Cisco's website, for example, of 107 different components they use and version numbers, often referred to as a blueprint for attackers, but they go a little further than they are obligated to do, but they are essentially already telling them on support documentation, not for defender use, but really just for license compliance. I try to break this down into a three-tiered, three columned thing. The first one is the ingredients list, which is I believe what today's about, inventory, parts, lists. It's really important for one [INAUDIBLE] and suppliers. Most of my software will not be consumed directly by a consumer. It's going to be consumed by a medical device manufacturer, and for them to be compliant with FDA requirements, I have to supply my subordinate partial SBoM so that they could provide their complete SBoM. This is a turtle stop stacked on turtles kind of a problem and one of the use cases should be those complex, multilevel SBoM type issues or supply chain type issues and the software supply chain assurance form that meets quarterly at Miter [PHONETIC] with DHS, DOD, GSA and others has been tackling this long, long time. There's a lot of knowledge here. This is also stolen and borrowed from Demming supply chain from Toyota in the 40s. This was done to make more profitable, more

reliable, high quality parts, not to make safer cars, just to make more profitable cars. Some of the concerns do creep in, and I share those concerns around when you start the map to column B, which is what is the known vulnerabilities associated with that inventory list, that's where you could get a signal-to-noise ratio issue. You could have customer complaints or questions, people expecting that they are all vulnerable, or that they want to see them all removed, and I think some education awareness working groups could really help here but essentially CVEs are not a great definitive list. It's pretty poor, especially on Open Source, to be frank. This should be treated as potentially exploitable vulnerabilities not definitely exploitable. During an active attack to answer am I affected or where am I affected in a hospital context, this is an incredibly reliable mapping to at least shortlist from thousands of devices down to dozens of devices that might need further scrutiny. And then third-party talent or third-party software could actually help with number three, which is which is the actual exploitable ones. Show some sort of substantiated claim that we are not using that. Control flow is one, as Chris Wysopal pointed out, but a lot of the high-profile attacks, the one that actually started me on the path of concern about third-party Open Source software supply chain was July 2013. It was an [INAUDIBLE] vulnerability. It didn't matter how you were using it. If you are using it, you were vulnerable, and it hit almost all the financial services companies. It was a real wake-up call for that sector. That was direct exploitation. There's also chaining attacks; there's also things like deserialization attacks. One of those deserialization attacks was the Java JBoss one that I briefly mentioned which was a single deserialization plot on a single JBoss library that had [INAUDIBLE] for six years. The FBI and InfraGard warned hospitals that they were being attacked. For anyone using this, they could not answer am I being affected and where am I affected. Could not, and; therefore, got hit anyhow. They shut down patient care for a week, diverted ambulances out of facilities, canceled patient procedures. It was pretty bad. And this is an isolated outage from [INAUDIBLE] It became a foundational trigger for our task force. I'm not going to tell you everything the task force found but in our comfortable truths [PHONETIC], we found 85% of hospitals don't have a single SISO or security personnel. They are defending really old stuff that's way past it's end of its life. They are over connected and reachable by the outside worlds, these

flat networks, which means a single [INAUDIBLE] a single vice can take out patient care, and in the case of Hallowed [PHONETIC] Presbyterian, they didn't, in the case of WannaCry they did, but more importantly a typical medical technology has over 1000 CVEs in it, and we already know CVEs is a pretty terrible blind spot. One that we cited had 1400. It only takes one to shut down the hospitals. These 5% numbers don't make me feel better. It takes one, and it's happened repeatedly. One of the things I hope Jennings talks about in his next one is during WannaCry, most of the hospital people I worked with on the task force had to call every single one of their vendors and beg to find out am I affected, am I affected? Do I need a patch? I'm blind. Help me out here. Couldn't get answers or got false answers. Incredibly inefficient both for the people at the vending [PHONETIC] hospitals and the manufacturers who had to answer a flood of calls inefficiently and inaccurately. This transparency could be lookups on, Oh we're not even using that library at all, let alone the wrong version. In the context of something like Hallowed Presbyterian, we are less concerned about even someone doing ransomware, just Internet noise can hurt things. I'm going to skip this, but we had a former Team Poison anonymous guy found the Cyber Caliphate. [PHONETIC] What do you think he would do with a known vulnerability that knows nobody can patch? In the last minutes here, I just want to point that people claim this hasn't been done, it can't be done. For about six years now, the financial services sector for the software they write and self-consume have been doing software billable materials mostly as a productivity boost. There's the entire rich market called Software Composition Analysis with many vendors and free and open source alternatives to those vendors. Typically, you will see 106 components on average, about 23% of them by volume have a known vulnerability associated, and even if one or two of them is vulnerable, those could lead to significant harm. The triggering point wasn't Heartbleed, it was actually prior. It was July 2013. Apache Struts hit most banks, huge wake-up call, so people started paying more attention to open season on open source. There has been a bunch of academic work. Dan Geer published something with me in 2014 in Usenix. There's been some economic analysis of that avoidable harm at the head waters [PHONETIC] at the manufacture of what is significant downstream breach costs, patent emergency patch costs, Frenzies. Underwriters Laboratories has taken on this with their cyber assurance

program which is now two or three years old. Several parts of the US government have touched on this and advocated for it including DHS through safety critical IoT principles. The Department of Transportation and the Auto-ISAC is fairly on board with this because it is very congruent with their normal physical supply chain management. The Presidential Commissions was calling for nutrition labels to enable more informed free-market choice, and the question is going to be, how much granularity. The White House said known but unmitigated vulnerabilities are among the highest risk to the federal government, and they are specifically referring to 10-year-old known vulnerabilities found frequently. Here is the Commissioner of the FDA has also been calling out and announced yesterday their intentions to roll this out at least for medical devices. My intention and hope is that this group can help make sure that that's a better program deployed earlier and more consistent across things because as a software producer, I don't want to have 10 different standards for 10 different sectors. Here is the lead up on the board, some workstreams you might want to consider, but I want this to be done better with the collective might and intelligence of this room. Thanks.
[APPLAUSE]
Last but not least, I think it's very important to get the cusp [PHONETIC]
where the user perspective of this data, so we have Jennings Aske who's the CISO and VP of New York Presbyterian Hospital.
No slides for me. I don't like slides; I am a lawyer by training, so I just prefer words. I am the CISO at New York Presbyterian Hospital. New York Presbyterian Hospital is a top 10 hospital in the United States. We're what's considered an academic medical center meaning we have university partners. For us, that's Weill Cornell Medicine and Columbia University College of Physicians and Surgeons. I am here really representing all of the healthcare providers that are dealing with this. In fact, this is a topic of conversation that we have with healthcare providers where it could be Stanford, it could be Mass General, it could be Mayo Clinic. We are all talking about this because it is something that does keep us up at night, so to speak, because we as an industry are arguably the most attacked industry right now from a security perspective. We really see SBoM as an important patient safety matter, and I will explain why in a few moments. Healthcare represents a sixth of the gross domestic product of the United

States. There is almost 6000 hospitals, there's over 900,000 physicians. Nurses are the single largest employee type in the United States, and when you think about a hospital like us, we have all these diverse partners. It's a really rich ecosystem of interconnections. As an example, we have kiosks at Walgreens that you can walk up to and get an urgent care visit. You can get your blood pressure taken, and we have to make sure those are secure and the software that's used in those doesn't actually pose risks to us. We develop telehealth apps ourselves. We do a lot of software developments, and we have to make sure that those are, in fact, secured just like the suppliers that we buy from. Now, healthcare truthfully is an industry that really hasn't done a good job of being secure. Healthcare kind of is taking some lumps and we are continuing to take them. I am happy to say that, though, it's getting better. You see hospitals like mine making investments in security at the same level that banks and other financial sector players do. I'm looking at my organization's security posture and where I see the greatest risk right now, it's with software. We've implemented many of the infrastructure controls like intrusion prevention. We've implemented privileged access management, but software remains our single biggest issue, and I'm talking about it not just what's deployed in our data centers, but also what's hosted for us. In fact, one example of this is that one of our electronic medical record vendors, they did not patch a third-party component of their infrastructure, and basically, we found ourselves unable to prescribe controlled substances. So, in New York State, we are required to prescribe controlled substances electronically. We do that to prevent opioid diversion or at least reduce the risk of it. For several days we could not because the vendor had not patched JBoss vulnerability, it was mentioned earlier. Healthcare is being hammered because of software vulnerabilities. Again, many of these are third-party parts of a product or platform. When I think about it and going back to an event with our EMR vendor, we purchase that vendor or the product in their services, and we basically weren't able to look into their black box because there is no SBoM, and at the time we weren't even basically thinking about this. This predates my time at NYP; it was probably about 10 years ago. The long and short of it is, one of the things that we are doing is starting to require our vendors to provide SBoMs. I'm actually working with a group called Viziant. Viziant is a group purchasing organization. It represents about

$100 billion worth of purchasing power across the United States. There's hospitals like Mayo Clinic, Mass General, New York Presbyterian that are part of this, and we are getting ready to actually finalize an SBoM that we will want people to submit. We view this as such importance for us to basically identify the 4%, 5% of issues that may affect us. I want to say as a customer, I recognize that all vulnerabilities are not exploitable. I'm not going to, because it's not in my interest in terms of my resources or yours, to basically chase down a resupplier and say, you have to patch this immediately. That's not just how the world works. The objections that are around the percentage of vulnerabilities or the difficulty of publishing them, they all strike me as kind of just false arguments. If you look at the United States, there is a history of industries objecting to implementing controls or best practices. I'll start with healthcare. Healthcare hasn't implemented security very well. We are trying to fix that. Healthcare also at one point didn't have fire safety as something that it took into consideration in terms of patient safety. It took two apocryphal fires, one in 1949 and one in when in 1950 to get hospitals to actually care about fire safety. Unfortunately, it led to a lot of regulation, and people sometimes focus on regulation as opposed to what's the best practice for this particular building. They'll say, I've got all the things that regulations require, I don't need to do more. I don't need to think about this intelligently and critically. And so, I'm hopeful that today and all these other parallel efforts, we can have conversations that aren't about regulation. It's about us thinking about how do we do this kind of intelligently, how do we pilot these concepts. I talked to Jim earlier today about possibly having some piloting of this with New York Presbyterian, some groups of healthcare providers that are purchasing products from Siemens is an example. I've talked to other medical device manufacturers. I mentioned the Vizient work where we have literally a group of hospitals that are all increasingly doing this trying to create a standard SBoM format that we would like to see people adopt. That's actually not a good idea. I want to see us have our own healthcare industry SBoM. I'd like to see us leverage something that becomes a standard, cross verticals that leverages SBoM standards that already exists like as SPDX or something. So that is where I am trying to steer things. But I really want to see this be a partnership, and I also want to mention about objections. As Josh said, this is already happening. If you sell to the defense industry,

you are providing this information and you are going further, and you are providing ports and protocols, configuration items. It goes well beyond what's in a typical software bill of materials. I think, also, one of the things that troubles me, and I'm not saying this to be antagonistic because I really want this to be a collaborative discussion, but there's always these comments about the customer maybe not doing the right thing with the information. These are the sort of objections that were made about seatbelts and in automobiles. Car manufacturers said if we put seatbelts in, people will drive more recklessly. That's insane, and in fact, we all know seatbelts are great. They are really wonderful things that save lives and the one thing the auto industry learned was that by implementing safety measures, that people would one, they would assume the cost. They had no problem with that because they valued the outcomes, and they actually saw that safety was a selling measure. Look at Volvo and others and all the things they are doing around safety as a way to selling to customers. I guess what I would say is, I'd like to see coming out of this room consensus that this is possible. We may not know exactly how but there is value here. There's value to you as a patient; there's value to you as a customer of other industries, and I really would like to see us move past the FUD and just focus on how we could do this in a way that's tenable by both sides of the coin, so to speak. This can't be something that's overwhelming, and I would like to see us do it before we get some regulations that no one's happy with. Thank you.

[APPLAUSE]

Thank you very much. I really appreciate that. For those of you who are curious, as long as it's okay with the speakers, I will put their slides online so that they could be seen as outcome [INAUDIBLE] facts and figures. Second, I just noticed that we're a little crammed by the entrance. We have some seats here. There's a little more space, and there's power and it's a lot easier to speak, so if you want to come down, don't be worry about getting in front of people. We're going to have lots of chance to talk, so we hope that you can come down if you want to make some room in the front. There is a lot of seats. This brief note on the schedule, something that I failed to mention earlier, is during lunch break, we are going to take a little bit of time to talk about another workstream that NTI and the Department of Commerce were in with our NIST

colleagues and with our DHS colleagues. The government published a report to the on president distributed automated attacks, slightly related to what we are talking about today that lays out some goals for addressing this ecosystem-wide problem, and we want to share where we are from, a roadmap of where we are going next in terms of developing a roadmap to make progress on the action items laid out in that report. So, I wanted to flag that for you. But now comes the hard work. We had a great range in perspectives. I won't recap, obviously, but I heard a number of things from different sides; which is one, acknowledging some of the challenges, understood the importance of having a shared vision across the ecosystem, and then seeing what we could do about it. So now is our chance -- on the schedule one of the things we flagged to get the ball rolling is to talk about what our potential use case is, so I thought we would start there, but if there are things you feel we should be putting on the table for perspective that weren't out there or things that we need to emphasize, now is the time to chime in. For those of you who are on the call or those of you watching the webcast, there is a challenge which is, due to the nature of webcasting, there is a slight lag between the webcast and the call. The call is much closer to live, it's a few seconds ahead. If you want to participate and join in that discussion, just make sure that as you integrate with the call is what you are doing you're going to get a little disconnect. [INAUDIBLE] Who has an idea? Who has a thought to share? You can hit the button that is right in front of you and that will light it up and get you in the cue. I will also keep an eye out for folks who are raining their hand who want to speak.

Good morning everyone. I'm Frank Giorno with Oracle. I think one of the concerns that I would have about when and how to do this transparency is, if you take Oracle as an example, we know we have hundreds of thousands of products, and we have tens of thousands of customers around the world; and each customer, obviously, there isn't just one sort of person who has some kind of like security credential who has been cleared who has some security clearance to have access to any information that we provide about our products and to maintenance, etc. How do we -- assuming for a second that the information about not just components, but vulnerable components and vulnerabilities in components that are actually exploitable, how do we provide that information essentially to the good guys, to our customers, who can do something with

that information without providing it to the rest of the world, to the bad guys? I mean, when you communicate to tens of thousands of individuals, you might as well put it on the front page of the Washington Post. How do we address that concern?
[INAUDIBLE]
JC Herz from Ion Channel, and we do software supply chain insurance in for national security customers. The notion that the bad guys don't already know is crazy talk. Like there is this weird notion that if by publishing something that you would somehow inform the bad guys because they don't already know, and that's just flat out wrong. The cost benefit of publishing vulnerabilities to defenders -- the defenders are the last to know. We just need to trash this idea that somehow the defenders are going to publish information that attackers don't already know because that is not constructive. Let's just assume everything is already compromised and then figure out what to do.
[INAUDIBLE] Financial Institution, I want to make sure we state it as a use case. Okay. To refract it as a use case, the use case is, I would like defenders to have the same information that attackers already have.
Can I just make a point on that? We do binary analysis to find the components. The way that we are doing it is to actually just look at the product itself. We don't necessarily have access to the source code when we determine what is in there so, of course, attackers can do that. The whole existence of all those CVEs out there, most of those are found by independent researchers who don't have access to what the vendors have access to. They are coming from the attacker perspective.
Constance, we have Kim Price on the call. Kim, you are going to speak to us from the ceiling.
Fantastic. I can be the voice from the sky. Can you hear me?
Yes, we can hear you.
Great. I would just suggest that because there is a lot of fear and concern, and this is a very complex scenario in terms of, Hey let's just start publishing all the vulnerabilities and all the components we use. I think potentially starting with a smaller and more easily contained scenario in use case in doing some proof of concept testing might be the way to go. If we can identify something that works in healthcare or automotive or critical infrastructure and see how that works, then we can look at expanding it more

broadly and have some wins and key learnings potentially from a smaller test.
I like that idea. Art.
Art Manion. I think I really do like the smaller scoped testing idea. One concern I have with that is, since a lot of the components we all use are very general-purpose, even a smaller scoped one would require the upstream components to be part of the system, so you may end up with eating a significant number of components already involved. For instance, I have specific healthcare software with JBoss in it, so you have to have JBoss, a bunch of libraries just to even do your smaller scale experiment. But I do like the small-scale experiment idea.
Eventually we will get somebody who hasn't already spoken in front of us, but please this is how to get the ball rolling.
I was going to mention that that small scale POC, to me, is the only way to move this forward. If we start talking about boiling the ocean and doing this across multiple industry sectors, it's not going to happen. I think maybe it has to happen with POC's in each vertical. Certainly, in healthcare this is the thing that I've been talking about. I mentioned during my presentation. I talked to several medical device manufacturers, GE, Smiths Medicals, others about possibly being part of this POC. I would love to evolve other stakeholders in this. I mean, basically New Your Presbyterian is offering to convene this process and bring people together.
That's great thank you. I think one of our approaches having this be sort of small localized is fantastic, but we want to make sure we are bringing in the voices as Art mentioned from broader space.
Just to clarify, you can do small with regard to participants even though your transitive dependency tree is quite large. I think there are -- we should leverage institutions for like the ISACs and industry groups and as much as possible be industry led because the last thing you want is a large, centralized bureaucracy kind of mandating that this is the place where you are going to come, so to the degree that we could leverage these individual groups and sectors and verticals that's amazing.
Michelle [INAUDIBLE] from Novalia [PHONETIC] I agree that starting small is a great thing, but I think that one of the requirements of that initial approach needs to be the need to be able to be scalable. Things like machine readability and consistent

terminology, naming conventions, so starting small is a great way to test it out, but that has to be in the idea that it's going to become larger and grow.

The question of small is not clear because people have different ideas in their mind. One of the problems we have is there's lots and lots of vulnerabilities out there and hardly any actually get exploited as a percentage basis. I don't know what the numbers are, but I would be surprised if it was more than 1%. An approach might be let's look at cases where actual exploits occurred and then work back from that and figure out what we could do with transparency that would have blocked that sooner or made it available. That's just one idea.

I just want to comment too, when I say small, what I am thinking about is something that would involve many of the large academic hospitals across the country like Mass General and others that really treat millions of patients, so this is not going to have a small impact. And to the comment about machine readability, absolutely. I mean I think as we -- I have a pretty large security team. We have 56 FTE's, probably about 10 consultants. That's pretty large for a hospital. We are spending over 10% of the IT budget, but most hospitals don't. They are starting to head in the right direction, and we don't want to deal with a thousand PDF's. We have across our six Manhattan hospitals just in the sense of medical devices, we have over 1400 different medical device manufacturers with 72,000 medical devices, I can't do the math in the number of PDFs. I think that aligns with the idea that we do have to think about what are the outcomes we are trying to achieve and how do we work back from that? I only know that I think that the fact that things aren't exploitable is maybe the outcome, but I do agree that we need this to be scalable because I can't have too much noise as well, so I want to echo that there's noise in both sides of the equation here that we have to think through.

Just to not to accept the assertion that this has to be tested before it proves its merit, it's been done at scale in financial service institutions for over six years. They are doing it for the software they write themselves and consume themselves, so they are very mature processes. Some of them are in the room today and can speak to how mature their processes are. It's not about will this work. To me it is a question of how do we make this traverse a commercial relationship where there is someone making out of station and consuming it in consistent ways across those. The practice exists, the

deployment modalities change.
Further thoughts.
This is Michelle again. One other use case to consider here is, it's not just knowing where the vulnerabilities are but where they aren't are. So, vulnerabilities that have been assessed and have been determined to be not exploitable on a certain system or in some situation, that A versus B that we are talking about, it's in A, but it's not necessarily vulnerable in B. I think that's a really important part too. We heard that a lot in [INAUDIBLE] when we were initially having a lot of those conversations with the hospitals is, I don't just want to know when you found something that's a problem, I also want to know it's not in these other products as well. I think that's another use case to make sure we consider.
Something that could incorporate the fact that we've settled this. Further thoughts.
We've talked about what a pilot might look like because we have a number of folks in healthcare, we've raised that. We've talked about financial sector as another area.
See I did learn to press the button. Open data, the importance of open data. Having been an Open Source software policy person in DoD, there's a lot of encumbrance that is created when people use a lot of [INAUDIBLE] tools where the information is encumbered by an electro [PHONETIC] property restrictions about how it's used or how it could be shared, and the sense of causing a lot of problems. The commons for sharing information needs to also encompass the right to share information that is unencumbered by an electro property restrictions so that this can all be done legally with respect to copyright because people often times want to share, they want to use, but their contracts are written so they can't, and that's a big problem.
[INAUDIBLE] here from [INAUDIBLE] A comment on the possibility of the software components details being dependent on the impact that we should be concerned about, say a healthcare portal that does something about a medical information transaction compared to a healthcare portal that just publicly available for everybody to see what New York Presbyterian does are two different things. Is there a reason for us to think about the impact also when you talk about use cases?
Classic, thinking through the risk as well as the [INAUDIBLE]
Is it possible? Are we will be able to do something [INAUDIBLE]

Sebastian Benthall NYU's Center for Cybersecurity and also Ion Channel. There was a certain amount of saying whatever NVD is doing, it's not good enough right now. I guess I've got a question which is why is that the case? To JC's point, if there's a commons of open data, there might be a tragedy of that commons, and I'm wondering whether from a policy standpoint there's things that could be done to change the incentive structure about contributing to that because my understanding is that they really depend on vendors for input.

Unfortunately, my commerce colleague from [INAUDIBLE] is in Montréal at the ITF meeting but, we do have Tom Malar [PHONETIC] from DHS whose been involved, so I'm going to give him a two-finger response to that.

Tom Malar from DHS. There is a little bit of, if people are familiar with the Mythical Man or Mythical Person-Month, there is a little bit of that where there's only so many resources that can be applied to the work of NVD, so we have encountered that over the years. So, I'm not even entirely sure if throwing more people, time, resources at the problem there starts to solve that. But focused use cases for that is probably one way forward where we could at least be investing in exactly the right things the community needs. I did want to bring up at least a couple of use cases from a couple of different perspectives that we see in DHS and that I've heard from other people, not only in this room but elsewhere. I think one is, where is the plimsoll line or the water line for what I don't have to tell you. If we want to have open data, that's great if your code is all GPL'd and you are using diagnostic tools in your toolchain, and as soon as you build the package it just tells you everything that is part of that package. And I think the Lenox Foundation is moving in that direction and that's nice. Whatever we come up with for critical infrastructure and other places where a mixture of open and proprietary software is used for really important stuff has to match what -- if the Lenox Foundation, if the core infrastructure and initiatives start to converge on a solution for their issues in this space and then another solution is arrived at by a non-interoperable SWID implementation that's better suited for proprietary software manufacturers, then everybody else's job just gets harder. A little bit of that is Josh's point earlier about don't have 10 standards for 10 sectors, so let's not have even two standards for two families of software. I would really like to get us to, how are we going to control the vocabulary so that at past a

certain point, whether it's GPL'd or not, you can just say -- Like you see on a food label, sometimes if you read the ingredients on a food label you will get to the part where it just says "spices" or on the back of a Coca-Cola bottle where it says "natural flavors." That's basically telling you there is some other stuff in this beside all the common stuff you already knew about, the water and the sugar, etc., but we're not going to tell you; that's our proprietary IP. We need to control that vocabulary. We need to determine where that water line is , we need interoperability with things coming out of purely Open World.

We've gotten to the portion where we are going to talk about standards.

Kent Landfield [PHONETIC] My real question here is scope. We need to start discussing the scope of what we are really trying to focus on here. Is the software bill of materials going to be as was mentioned earlier code snippets? I think that's probably out of the realm of possibility. Is it third party components that are not developed by that vendor? I think that's a good place to start. Our real focus here is that we need to establish a scope that is workable for a lot of different sectors and beneficial to the customers we are trying to serve. Right now, I don't -- and I am a little confused with the focus on intellectual property copyright issues. When it comes to Open Source software today, they demand that you as a vendor put their disclaimers in your products that you are using that. It's in a PDF somewhere. It's in a document somewhere. I'm a little confused as to where the intellectual property rights for naming a specific component that you already have to declare in your product comes into play.

Question about the CVE. I'll acknowledge there is a CVE sort of coverage and counting issue, and I think a lot of the folks in the room are aware of this; however, it is a bit separate from I think the SBoM discussion. So, I would have no problem with a supply chain transparency SBoM initiative going forward based on NVD or CVE data. That's enough to certainly get started. Separate question about are all of the vulnerabilities in CVE, interesting separate question, but I think it's separate enough. It won't hold us up here.

Constance, we have Brad Ree on the line.

Hi there. I'm VP of IOT Security at Verimatrix, but actually I will put on my different hat, which I am also the Security Chair for the Zigbee Alliance, and for those who don't know

Zigbee is really focused on basically smart energy, smart building, and smart homes. One thing that we actually already have as part of our certification compliance kind of program is, our device manufacturers are required to say what platforms their on. So, we are already sort of doing this bill of materials, offer a bill of material, but we actually don't have it connected to vulnerability which would be great to have in there. I do think as we started talking about it, there was some talk about group of concept that writing off of -- some of these other standard bodies I know Open Connectivity Foundation is doing similar kind of things with saying what platforms you're on, but the part that I am a little bit nervous about is who should actually be seeing the vulnerability? Should it be the manufacturer of the device in Zigbee speak, Hey the platform you're on has a vulnerability, please fix it. Or should it go all the way out to the end customer, which really could create one, a lot of noise; two, it could create supply chain problems as the manufacturers try to address this stuff, roll it on out, addressing the products in field and when those will receive the updates. Really, the two parts being, one, can we look at what standards, organizations to actually work with that might have some of these pieces in place; and two, my other question is, who should see the vulnerabilities and how should people be able to subscribe to those. So that's sort of my concern there.
I want to -- it's the moderators prerogative first, to carve off a couple of things as Art said, this is -- we are touching on a bunch of different related issues around CVE and vulnerability disclosure. On that latter topic, on vulnerability disclosure, many of you in the room worked through an NTIA and produced some wonderful documents to talk about this and have been carried on by a lot of companies that are represented in the room and organizations that are in the room, so I want to sort of leave the vulnerability question aside unless there's something that we want to pick up. Similarly, questions around CVE and NVD are very important. There are a bunch of discussions going on, again, led by many of you in this room, were involved with many in this room, and we should definitely draw in and out of them, but I don't think we want to zero in on those particular topics.
Could I just add a comment to that? If you have a set -- you have vulnerability information out there, maybe it's in CVE, maybe it's in other databases, and then you have a list of SBoMs out there, when a new vulnerability is disclosed in a component,

then all the products that use that, it's sort of instant disclosure, right? Like this idea that there's unseparate disclosure from the people that are the consumers of the component doesn't make sense to me. If you are publishing your bill of materials, and there is a vulnerability disclosed one of those things, it is disclosed, right? There's no separate disclosure. The manufacturer might put out a separate security bulletin, but it's public information that these two things are public. It's adjoined of those two data sources. But the fact, as other people have brought up already, I think somebody said that 5% of the vulnerabilities in a component are exploitable in the product. [LAUGHTER] Numbers like that. The problem we have is, I don't want to spend all of my time on the one out twenty -- I want to spend my time on the one out of twenty that actually are exploitable, and we need to focus on that because if you spend your time on the nineteen out of twenty, that doesn't do anybody any good at all, and it prevents them from applying real fixes that are needed because people just can't deal with that noise. Brian Fox from Sonatype. Just a follow-up on what you said, I think if you are doing a good job internally you've already investigated those other 19 internally and decided amongst yourselves that that's not exploitable, so, therefore, then going ahead and publishing that information to your consumer seems like a really good practice to do. I don't think it's actually extra work; it's probably work you're already doing internally. If the experience with the common vulnerability scoring system has taught us much over the past half-decade or so, I feel like the problem is that the 20% -- we are pretty sure these are the characteristics of that 20% or 5% or 1% are what the things we need to watch out for, and we can do a 10 year research project and involve lots of universities and lots of really smart people, and we will still get it wrong because we will have successfully fought the last war, right? I was pretty sure buying processors from Intel wasn't going to affect the security of my cloud environment and I was wrong. We all were. We didn't do much else, and that's not to drag any particular vendor under the bus. I want to be very careful about that because I'm just saying -- I'm point using them as an example to point out that we are all going to make mistakes in that area. So it is extremely difficult, and I am worried that it is possibly one of those long tailed, world-class, due 20 years of R&D and still get sort of a maybe result research problems to determine and predict which are the X number minority of vulnerabilities that are likely

to be actually exploited because we thought RCE was the only thing we had to worry about, and now we live in a different a very different world.
Katie then Jennings.
Hi. Katie Moussouris, CEO of Luta Security. When I was at Microsoft a decade ago, they launched something called the Exploitability Index. It was in addition to the Microsoft Bulletins. It was the first major revision of the Microsoft Bulletins in many, many years, and the idea was to actually predict which of the vulnerabilities in the Microsoft Bulletins were most likely to have reliable exploit code available within two weeks. One, the Exploitability Index was very difficult to educate consumers of the bulletins on how to use them. They were already using the existing taxonomy to decide which of the patches to apply in which order, so they were just going by the criticality rating of Microsoft. Even when we pushed a massive amount of education towards the customers how to consume this data, the Exploitability Index, it still was just a window of time that we were really trying to do this prediction. It was really about patch prioritization. So, even if we produce it, we already have evidence that the actual enterprise and all the way down to the end user consumers don't actually use this information. They still go by the criticality rating overall.
I just want to comment about a little bit of [INAUDIBLE] hearing some comments which almost feels paternalistic in the sense that you won't -- we as a consumer, so to speak, of products. We buy these products, and the last thing I want is for the vendor to say, You can trust me. I don't mean that to be disrespectful or anything, but vendors make mistakes. We make mistakes; everyone makes mistakes. That's part of the human life. I can count on more than two hands the number of times that vendors have said, You won't be affected by this. And you will, or we were. WannaCry, it's one of those things that we had pharmacy cabinets that were affected by WannaCry. We had to shut down clinics because vendors didn't patch and things like this. The whole idea that this thing may not be exploitable today doesn't mean that it won't be, and also, your assumptions about why it may not be exploitable actually may not be accurate in the environment in which it's deployed. You can't account for every single entity that is deployed your product whether it's the flatness of your network or maybe they've just misconfigured the device, and there is a port [PHONETIC] that's open that shouldn't be that actually

makes it. Exploitable there's a reason that a hospital like mine and others care so much about this is because we want to play a role in our security posture, and we need you as suppliers to work with us, and I get that there are burdens and costs with this, so let's try to figure out something. I want to go back to the concept of a POC. I can envision a scenario where we have some critical healthcare suppliers working with folks from different -- stakeholders in this room on defining a POC where we look at things like, how does this information get conveyed to the customer? How does the customer leverage it, and how does the customer and the supplier engage in feedback over the course of this? And try to learn lessons that can be used by other industry verticals as they think about this.

Two fingers from Josh, and then we are going to go to Kim on the phone.

We have somebody from FDA here, granted it is a different part of FDA, but I'm constantly perplexed by like, they must have had almost identical debates over the nutrition labels in ingredients list, and the truth is most people ignore these and don't know what half the ingredients mean. But if you do have a health issue and your doctor says, Look out for your intake of XYZ, you can. If you have a nut allergy you can avoid it, if you're gluten intolerant you can avoid it, and it doesn't preclude you from selling junk food or buying and consuming junk food. To foot stomp this, we had severe pushback in the healthcare task force from people feeling like they were being talked down to or infantilize that they couldn't possibly handle all of this, or they are going to get it wrong, or they are going to ignore it. I really feel, to advocate for them, that's their choice and we need to equip that choice because the ultimate risk decision is on the operator, and if they are denied the information to do that risk assessment, then we haven't equipped them. They could ignore it; they could do it badly, but we have to equip them to do so.

Kim

The concept of only 1% or 5% of these vulnerabilities is exploitable keeps coming up, and I would like to point out that for most, not all, but for most Open Source and third party components, when they release an update it includes multiple fixes, and we are not suggesting, or at least I hope we are not suggesting, that the consumers of those components go and point fix specific CVEs. I think that we are talking about taking a

component update that fixes multiple CVEs, some of which might be exploitable. When we come down to talking about the scale and how we can't patch everything, it's that argument of, well, if I have to prove what's exploitable to get you to update, you probably could have updated the component in less time. So, having that hygiene conversation, it's very much a culture change for those developers using Open Source and third-party components and so, I think it applies here as well when we are talking about the scale of things. It's just to keep in mind we are not talking about thousands and thousands of fixes. We are talking about a smaller number, still a substantial number, but a smaller number of component updates that fix multiple things.

Thank you. JC

I think a lot of this points to the business process issues behind when we say things like transparency and code. Remember way back when we had Internet connected TVs that were an updatable, and we all kind of decided that was a bad thing, and so having un-updatable components was declared to be bad. That's a business process decision that might be different in different sectors, but when we get to these business process issues like, is something automatable, right, so that you can scale it. Are machines readable so that you can have API calls? In some ways, the resilience issue is more important than the vulnerability issue because if you build in the business process that enables resilience, you are better positioned to address the vulnerability, and nobody wants to talk about business process because it's hard. Nobody wants to change their workflow even if it improves their workflow. People hate to change their workflow. But some of these best practices, my gut, and from what I've seen in some of these large enterprises is that a big component of best practice has to do with business processes that address resilience, not just the what's in it or the detection of the vulnerability or even how we name it, all of which is necessary.

Just to echo that, I think one of the things that we have observed is that when you can easily produce an SBoM, we know the team that is producing that software is highly efficient, and it's not necessarily because they were trying to produce an SBoM, it's because the processes that undergird their ability to create an SBoM are already embedded in their practices. The cause and effect, let's make sure that it's clear, we are not driving an SBoM so that they have better practices, they already have better

practices and an SBoM is a symptom of that. That's why we want to be able to highlight. One other data point, I think there's very few things new under the sun. We should learn from best practices and other industries and other sort of things. There is something that we are all vulnerable to, believe it or not. It's called something like pandemic flu or Ebola. Would we criticize the government if we said, Hey you're not being transparent about the presence of this new flu variant? I think we would want to have the government tell us, Hey, there's this new flu variant going around, and maybe you guys should all take action. As adults, we should take action. It's our own decision to take action such as getting a flu shot, but if you didn't get a flu shot, then that's on you, but also more importantly, we do want a herd effect, which is we want enough people to get flu shots such that it doesn't cause other collateral effects on other people. Why do organizations want our employees to get flu shots? Because we don't want them to get sick. That reduces productivity. Ultimately, this gets down to those sort of core issues. We want people to be productive; we want people to be efficient in doing what they need to do. Sure, it helps us with security too, but one of the core issues is us being more productive, and when it comes to software bill of materials, again, that is a side effect of being more -- having a better process for how you do development.
Hi. Mark Willis from Aetna. What I am hearing here, I am just concurring with a few things that we definitely want the defenders to have the same level of knowledge as the attackers. This gentleman in front of us talked about doing the right thing. I think it comes down to the word "responsibility." We have a responsibility for those of us that write code and produce software to the public for use. It comes down to, are we responsible if we know that there are vulnerabilities that may be exploitable or may not. Again, as the gentleman from New York Presbyterian Hospital, Jennings, what's exploitable today or not exploitable may be exploitable tomorrow. We know this for a fact, and so to wait until the car is off the assembly line to find a defect down the road and do a recall, I think it is not a very responsible way of looking at things. I think if we have the opportunity to, and it sounds like we have enough people in the room that have the experience to put together, like the gentleman in front of me said, about a single standard, I think that's what we should be aiming for, that single standard to be able to build consensus. Again, I want to focus on the word "responsibility" because we

are kind of touching on dancing around that, and that's really what it comes down to. Are we being as responsible as we can to the public and for those who use our software? And if not, we definitely need to keep that in mind as we work on that single standard and try to make this better.

Thank you. I've got one and then we are going to go to Steve on the phone.

Michelle. Michelle [INAUDIBLE] again from Novalia. I completely concur that good business process and good design controls are essential. I think that a lot of the folks, and I have been involved in this discussion from healthcare side for quite a number of years now, I think folks get that. I think one of the bigger challenges, though, is the large number of legacy devices that are out there that there isn't a way to upgrade them because they are in old operating systems that would require a large amount of hardware updates to get them to work. So, I think one other use case to consider here is to make sure that whatever we put in place can also safely be used with older products that cannot be updated due to the component manufacturer's no longer in business, the operating system is out of date, because we know that that's not going to be a problem that's going to be solved anytime soon is the large number of legacy devices out there.

Josh very quick two finger.

Very quick. We spent months on this on the Congressional Task Force for Healthcare. One of the nice things about an SBoM is it can be done retroactively at least through impact analysis even if you can't go rearchitect the stuff, so FDA is trying to make new stuff, have a higher standard of care for hygiene, but enumerating the old stuff may be your only defense.

That's a great point. Steve Lipner from SAFECode.

Thanks. I think somebody mentioned a couple times that the financial sector has standardized on requiring an SBoM for both their in-house and acquired code, and one of the things I would like to hear is what their experience is in terms of making use of that bill of materials in terms of responding to vulnerabilities, actually taking action to use the information they've got. If somebody is there and could speak to that, somebody from the financial sector is there and can speak to that experience, I think that would be informative.

As it happens, we do have someone. Thank you. That's a great thing to put on the table.

The specific reference in terms of us standardizing it is based off a [INAUDIBLE] that we wrote in 2015 that lays out what our expectations are and, it includes contract language even to say this is what we would want to ask of our software providers. Now, it's up to each individual institution to implement that. That's not done consistently across the board, but we are trying to get there. I, for one, have actually introduced it into language when I do transactions with certain software vendors. Again, it's not consistent across the board, but the point being that we are moving towards that. What's the response? Well, it turns out that most -- I deal with a lot new of startups, new companies, and funny enough, they actually have pretty good software development practices, and so creating an SBoM is trivially easily for them. One ideal way of some of the more legacy vendors, it's a little harder as one can expect. I think this actually points to a specific use case, however. The use case here is this: If a vendor can tell me what is in their software, then I make a certain risk decision based on the knowledge of that. If they cannot tell me, it becomes a negotiation point for me because I have to then spend additional time and energy and money to understand what software I have in my environment should an incident occur, and I want to have them make concessions upfront because they are going to cost me money downstream. When an event like struts [PHONETIC] comes around, and I need to quickly enumerate where do I have struts in my environment, I have those who comply, and I save tremendous amount of time knowing that upfront, but for those vendors that don't, I have to spend a lot of energy to figure that out. The negotiation point here is this: Since you cannot tell me that, then my negotiations, I want you to make certain concessions, and that can be in whatever form it needs to be.

So, I heard two underlining questions in Steve's point asking for some response, and I think we can hear from financial or from healthcare or any other sector that might be here. One, is just the standards question which is, did you guys come up with a format? Is there a lesson that we can take away there? And then, two, other than your having this as your high-level risk contract, are there lessons that have been learned about how this data is integrated into your day-to-day operational and ideally automated security

decisions? I'll give you the floor, but I think that might be something that other people can jump in as well.

Brian Fox from Sonatype. We have been helping companies do this for close to seven years now. A lot of them in the financial industry. A couple of antidotes come to mind. I spoke to one large bank after the Commons collections issue, the one that got Hollywood Presbyterian attacked. They told me that they spent three months with four people so a man year [PHONETIC] just going around the organization inventory and where they may be affected by that particular vulnerability versus other financial industries who had already been using software to know definitively where it was, they instantly had that answer. In many cases, even before the data hit NVD and was public because the information about that was out in various channels before it was officially published in NVD. There's the tale of two cities, if you will, the huge discrepancy in the response time. Additionally, we see many of our customers using the software not only to improve their internal practices where we see best cases leading to 30% increase in efficiency by companies that are doing it, but they are using the software to analyze things they are buying from other people. Which means just because you don't provide the bill of materials, doesn't mean your customers aren't already doing it and managing it on their own because I think we all know these components, while their published, we are fond of saying they age like milk not like wine. It's a case of when there is going to be a problem, not if, and how are you going to be able to know how to respond.

One really quick metric and just to echo this because we do the same kind of processes as Sonatype, and lead qualification, and lead qualification to your point about the BoM, we say to an enterprise to assess whether they are actually ready to do security automation for continuous integration, Okay, there is a vulnerability announced against some software component. All I need you to tell me is the unit of time that it would take you to know whether it is in your enterprise. Is that seconds, minutes, hours, days, weeks, months, or years? The rough order of magnitude, interestingly, it's like the pain scale 1 to 10, they can usually tell you, and it's a pretty good metric to start with. Driving that order of magnitude to assay down from months to weeks, days, or hours, that's definitely a skill scaffold you want, and there are techniques that help you get there.

Sort of on that note, just the vendor asserting that they have an SBoM and they have

that internal process gets us a long way to the end result we want which is secure products. I would rather have a vendor that had the SBoM and had a process for dealing with new vulnerabilities that come out in the components that they are using on a quick basis then one that just merely publishes their SBoM. I want a secure product. The SBoM is a means to that. Making it public, does that make it -- is that better or is it better if they can assert that they can deal with the SBoM quickly?

Getting back to use cases, I would like to distinguish between software products, which can be delivered, and the consumer of that product can run certain scans on autopilot themselves, and devices where it tends to be more closed and harder to do that, or sometimes even impossible to do that.

That's a good distinction. I noticed that we have been using this term "software bill of materials." Do we think that there is a shared vision or that we can come up with a shared vision of what that term means that captures most of the space we are looking for?

I will get back to the question of scope, what is it that we're going to be trying to target in a software bill of materials? We have two institutions here that have mentioned they are either at or near having a finalized version of the software bill of materials and the machinery that will form. What are you using and what is the critical information inside of that that would be different from something like a software ID tag that has a lot of that foundational material already there?

I guess that question is for Sneal [PHONETIC] or Jennings.

We actually have -- we've looked at SWID. What we tend to find is, because SWID was created within the commercial software industry, it's primary, well not primary, but one of its good purposes is that it allows enterprises to count up the number of software units they are using so that they can pay the vendor appropriately, and it's good for entitlements. The problem with SWID is that the vast majority of Open Source doesn't even know what SWID is, and they don't need no stinking SWID tag. So, it's not something relevant if you're not getting paid. Now, there's all sorts of crazy proposals to like make them do SWID and then pay them through Blockchain [PHONETIC]

like there's some amazing ideas floating around. It's a little bit like in healthcare when you try to do analytics using the billing records and then you realize that a billing code is

not a diagnosis and that the diagnostic codes is sort of gain for billing, I know it's shocking, but it happens. So, things that are designed to get people paid, when you go into ecosystems where people aren't getting paid, you have a relevancy shortfall. We sort of need to recognize that, and we don't know what the answer is, but to make sure that there is some kind of open standard that can either be pulled from projects or even projected onto them.

Currently, SWID is being used by a lot of commercial vendors. No question about it. There's also Open Source efforts to implement SWID and create foundational cases for SWID for Open Source. So, that part of it I'm not sure really is not a big deal. There's two major formats right now, SPDX and SWID, and both of them developed in different areas and both have their strengths and weaknesses. One is an international standard and the other is a Open Source project. The question is from a purchasing perspective and a procurement perspective, you are buying products, so there is a commercial aspect to this. The real question for me is, we need to come down off of the worry of what format. We need to figure out what format would be best for us to actually standardize on and then start doing some of these proof of concepts in these implementations that we can then vent our ideas on. SWID is already in place, Microsoft and quite a few other organizations already have it. It's not something that's unknown. You may already be using it. The question is, is it right the right standard for us to be using? Right now, it has real implementations out there.

I have two people trying to throw interrupts, so if it's okay, I'm going to go Tom and then Rob.

I actually, full disclosure, met with members of the Lenox Foundation this morning to talk because they were like, Oh, we're doing a big briefing on SPDX. I was like, Oh, are you going to be at the software component transparency thing? They said, what is that? [LAUGHTER] So, I am going to get them on Allan's mailing list, so that's going to be get squared away, but the issue that it presents is awesome -- this is even funnier since they bought Github, but if Microsoft is using SWID and the core infrastructure initiative is pushing towards SPDX, do we really have to have this culture clash one more time? Can we head this off at the pass and get convergence so that SPDX works inner operatively [PHONETIC] with SWID? And if that's something we could have a do

out from today's ramblings, then I would be very happy.
Only if we all agree to use [INAUDIBLE]
[LAUGHTER]
I have been doing development for both Open Source and Close Source for a couple decades, and one of the problem points here is knowing what subcomponents are in an SBoM. When you are doing it at a license level, what you do for commercial software, you just list the licenses you have and that's at the level you've done something. When you go into Open Source, that no longer works because you have a Lenox distribution that's all GPL, so there's one license, yet there's a hundred thousand subcomponents in there, and then you are trying to track version numbers, and a lot of the component -- you've got the package managers today that tell you what package depends upon what package depends upon what package, but those version numbers are fairly basic because they allow you to update a version without breaking a dependency, so you don't actually know all the sub versions you just know some of the major versions. Then there's the problem of when the software gets old like in the IOT space when you are using old Linux kernel versions in old software of everything and they are no longer supported. You're even off the LTS, long-term support for these packages, so the vendor of the device -- if you take a Cisco Linksys device, that's like 10 years old now, and they are still supporting it, yet, they have to back port all these patches for these vulnerabilities back onto that. This sort of information is far too granular to even get close to being trackable with SWID's. For example, they are not even canonical. Two different vendors might have the same back ported batch, but there is no common name for it so you can't tell. One specific example of this would be the Shellshock vulnerability that was in the bash of -- shell interpreter comes with Linux, but most IOP devices have Busybox, some have Bash, some have both. So, no amount of SBoMs would tell you whether you are vulnerable to the bash bug or not because that information was never tracked at a level that you are talking about -- you're hoping your model for an SBoM wouldn't be that granular.
Josh and Brian thank you. Do you have a direct response to Rob?
Rob brings some up some interesting stuff. We talked about some of these. I do think scope is a killer question and one that should get a working group or two. A positive

suggestion towards that is, we sometimes talk about this in the future maybe we could try something, I think there is a bunch of de facto outputs. One of the most common injects for a bill of materials you produce is by the build process, like your Jenkins build [PHONETIC] You can't build it without building it from parts that got pulled in. If you don't have the part, build breaks, so they are emergent practices. They may not be ideal or optimal, but there are a series of them, and I'm hoping one of the projects captures the existing observable practices and looks for best and worst and weak. Some languages like Java are very, very solid, deterministic coordinate system with GAB; some of them are pretty loosey-goosey, come and go as do please, and they will use the same names for different content. But I want to capture those and look where the 80/20 rule is much like the food. They do have the everything else or natural flavorings type stuff, but I don't think we have to hope we can figure this out in the next couple of years. I think we can observe years of behavior first. I have an uncomfortable truth to bring up, but I am going to hold it off.

All right. I think we know Open Source developers tend to be anti-establishment by definition, which means efforts like SWID are almost guaranteed not to be adopted within that community. SPDX also has been around for a long time, not really adopted within the Open Source community, but what's interesting is almost all of the modern components that we are talking about these days come with a native package manager. Maven, NuGet, NPM, Ruby has one, PiePie [PHONETIC] [INAUDIBLE] Docker.

They all have a native coordinate system. To participant in that ecosystem, you have to give yourself a name and make yourself addressable within the address space. Efforts outside of that inherently require somebody to make a mapping between them. This is one of the fundamental challenges with the CPE because, for example, that can't address the submodule [PHONETIC] issue. The coordinate system is not robust enough to map all of those cases. There is already an Open Source project called package URL or PURL [PHONETIC] that is making an attempt to actually allow all of these native coordinate systems to interoperate [PHONETIC] under one umbrella, and in my opinion, that's probably the only way to be able to get that to work for the rest of us from the outside because we are not asking them to take an extra step. It's a step they are already doing simply to exist in their own ecosystem.

Responses, thoughts? So, we are going to break slightly early for lunch, but first Katie.
I keep hearing the comparison to ingredients in food, and I want to say that what I think we are missing here about that analogy is that it would be more analogous to labeling every package of ground beef with exactly which cows went into producing that. It's not a perfect analogy to use the ingredients in food when we are talking about the technical undertaking, and I think Rob's points, especially about the complexity, I just want to make sure that we are not handwaving over the complexity of producing this type of work.
But we actually do want to know what farm it came from.
It's not the farm.
I was going to say basically the same thing. Isn't that we expect in our food that when something happens we know exactly where it came from?
The analogy fights are always fun when we try to figure out [INAUDIBLE]
[LAUGHTER]
I'm just saying that all of my concerns about this are around scope and scale, about the level of effort to produce this material, and the utility of the material once produced, and especially if you look at it from an overall ecosystem perspective, what resources are we then going to be devoting to consuming and acting on this particular type of consumable material about vulnerabilities when we already have an epidemic of being unable to address the vulnerabilities and apply patches in enterprises today for the materials that we already have.
Agree. I think that, again, the business process is that if you are not automating [PHONETIC] this stuff, if you have to have teams of analysts and humans dealing with it, you are already screwed. You are done, you are dead. I think that one of the questions that we encounter is this transition almost like from an [INAUDIBLE] to an industrial process. To throw another analogy into the mix, we talk about chemical engineering. It's just the whole factor is the process and its sort of temperatures and concentrations. If this is still being addressed with the same manual, analytic, and approval process, then, yes, there's just no way, and so it begs these questions about automation and machine readability and service calls and all of that business process.
We set ourselves up for a good afternoon discussion. I have said these processes are

messy. This is how we get some very good progress moving forward. One of the outcomes from today that we might want to start thinking about after lunch is tackling head on this scope issue. One of the ways that these processes have traditionally worked is with the use of targeted working groups that can further define and constrain the work they are focusing on. They work in parallel and complementary approaches. That's one way to do it, certainly not the only way we have to do it. We also could have people go off and draft things and bring them back, but that's something just to keep in mind in addition to talking about your favorite metaphor. Josh is trying to grab my attention for one last word, so we will give you the last word before we break for lunch. I wasn't trying to get the last word. I think it is absolutely true that there are squishy, nondeterministic coordinates like both Rob and Katie have talked, about, and we should flag which categories most likely fall in that squishy, harder one. It is also true that most of these high-profile breaches were the same vulnerability hitting the entire financial services sector or the entire class break on healthcare, and we should enumerate those as well. Back to the targeting of where optimal scope is, it's which ones are going to be valuable in most cases and which ones will be harder and should fall outside of this. But to state the uncomfortable truth that I omitted before, I think Simial [PHONETIC] said it best, newer, smaller [INAUDIBLE] CIDC shops do this as a matter of productivity and profitability, and it's very easy for them to do, relatively easy. It's not to oversimplify the legacy problem. I have a bunch of legacy code too, so I am saying it for my own purposes. When I showed this to my developers to channel something I heard from someone else in this room, they gulped. They were like, Ugh. And why is that? We didn't know better, we never managed this, we never measured this. A lot of those products, people have moved on. One of the reasons -- I love the [INAUDIBLE] Most of the libertarian [INAUDIBLE] S-Type people who are pro transparency, pro information, pro data are so terrified of being anything but opaque. One of the reasons I'm going to state for myself and some of the shareholders that I work with is, they are afraid of their past sins. They are afraid of either license violations they have been cavalier about or known vulnerabilities. Basically, when the developers gulped, they said, It's the right thing to do, it's going to take us some time. Everybody poops, everybody's bad, [PHONETIC] everybody's got some legacy sins. We all have to get

better. The consequences of failure are higher. So, I think part of the schism here is new software versus legacy software, and we should be very considerate and realistic about legacy with the right time horizons, with the right expectations, with when this stuff goes into effect and like I said HT and HTS are doing it doing this. If it goes well, it goes into effect. The sort of Damocles [PHONETIC] here is, like I said, HHS and FDA are doing this. If it goes well, it will go widescreen. Why don't we get prepared before we are forced to do it by leaning into this?

All right. With that note, with one extra unsettled metaphor added to our pile, we are going to take a break for lunch. For those of you who are watching along and are interested in the commerce work on distributed automated attacks tax and botnets. That includes a strong IOT. We are going to be talking about that here at about 12:35, so that should be enough time for folks to go grab a sandwich nearby and come back. Alternatively, if that's not interesting to you, please continue the conversation, and come back with brilliant ideas that we can all use to solve this. We will reconvene the multi-stakeholder process at 1:15. Thank you.

***

***