SINGAPORE – DNSSEC for Everybody: A Beginner's Guide
Monday, February 9, 2015 – 17:00 to 18:30
ICANN – Singapore, Singapore

JULIE: Welcome, everyone, to the session DNSSEC for Everybody: A Beginner's Session. We'll start in just a few minutes. Please come in. I would urge you to take seats at this table. This is a very interactive session. We really want people to participate. We're not going to have enough seats if people don't sit at the table, because I think we're going to be at capacity in this room. So as you come in, please do come on up to the table. We promise we won't bite you . . . much.

DAN YORK: All right, good morning. Yeah, good morning – good afternoon! As people check their watches. That was just a check to see if you are awake. Can you hear me? Sounds good.

We do have remote attendees. Hi, I'm Dan York. I'll be the MC and leading us through this discussion today. We do have some remote attendees, so we are going to ask you, if you have questions, to speak at one of the microphones. We have these funky microphones along here where we push the button and you can talk into little things over here.

Those of you who are up here, there are a couple more seats. As Julie said, we don't bite, but we would encourage you to come up here and fill in a couple of the spots here. We do have that. We also have this microphone, which I can pass around as well.

Let me begin by just saying welcome here to our DNSSEC For Everyone. How many of you have had any exposure to DNSSEC before today? All right, a number of people. How many of you have signed your zones? Not you guys over here. Okay. How many of you have a DNSSEC resolver that validates or a DNSSEC-validating resolver? Okay, a few people. Okay, good.

We're going to talk a bit about how this all work. The agenda is this that you'll see. I'm going to talk a little bit about what are some of the basics of DNSSEC and how it works. Then we do have a skit that we will bring you. A skit in multiple parts. You can see some people over here getting their t-shirts on and getting ready. We do do this as a way of helping illustrate how DNS works, and then how DNSSEC comes into play with that. We hope we'll give you something that's a little bit entertaining as well, as you get to see some people act out. We did actually have a rehearsal today, so we may be a little bit more organized. Well, we're going to try.

Part of the point of this is we're often very serious about all this stuff, so we want to try to lighten it up and make it a little bit more interesting and fun. We would ask you, think about your questions. One of the great times we've had at this session is we try to leave a lot of time at the end to talk about questions and to go through and really get some good dialogue going. We've had some very good and lengthy sessions around that in this.

I want to begin by stepping back and going back a long time ago to the very beginnings of DNSSEC, you could sort of say. I want to introduce you to Ogwina. She lives in a cave, as you can see, on one side of the

Grand Canyon. This is Og. He lives on the other side of the Grand Canyon. They would like to go and communicate because it's a very long way for them to get down. They don't get to talk to each other very often. They have to climb down one side and go to the other.

On one of the times that they are together, they notice the smoked coming up from Og's fire and they say, "Oh, let's go and start using smoke signals to go from one side to the other." They do this as a mechanism to go and exchange communication. They're able to go and talk to each other this way.

But then, there's a slight bit of a problem. Somebody moves in next door to Og, and the mischievous caveman Kaminsky decides to start sending his own smoke signals, too.

Our poor Ogwina on the other side doesn't know which smoke to believe. Which is the right communication stream? Which is the right one that she should be looking at?

So, she decides to go over to the other side and figure out who she should be talking to. While she's there, Ogwina and Og talk to the wise village elders, one of whom is caveman Diffy, who says, "I've got this idea." He goes into the back of Og's cave and he goes in there and he notices that there's a pile of strangely covered sand that belongs in just that one cave. It's only in that particular one. He comes out with that sand, throws it in the fire, and all of a sudden, the smoke is blue.

Well, what happens now is that Ogwina and Og can chat because she knows that only out of Og's cave can there get to be this special

powder, this special sand, that turns his smoke blue. Nobody else. Kaminsky? Sorry he can't get in there. It's all about this blue smoke.

That's a bit about what we're going to talk about. Blue smoke is creating this special way that somebody can know that the information that they're getting is, in fact, coming from the one and only person. That is what DNSSEC is all about is providing the knowledge that the information that you're getting out of DNS is the same information that somebody put into it. That's what it's really all about here today as we talk about this. We'll come back to our little story in a few minutes.

In the meantime, if we think again about how we have this very high level view of DNS, very often we have a picture like this. The root of DNS, then all of the various different top-level domains and then the second-level domains that go underneath that. This is a structure that we've all seen. How many of you have seen pictures like this for DNS? Pretty much everybody. This is the standard kind of picture we have in some flavor of that. But it all starts with the root, goes to the TLDs, comes on down to the second level and can go on down from there.

We have these words, these concepts, about the resolver who knows how to get to that root, is able to go down their each level, tells the resolver to go to the next one, until finally we get to the right answer out of all this – that process that we go through.

The challenge that we're here to talk about is the fact that there's no security in DNS. We're going to see very easily when we demonstrate this in a couple minutes how easy it is for somebody to spoof an answer and talk about that. We're going to go right into that right now. Wake up.

We have a bit of a little skit here to demonstrate this. I'm going to turn the mic over to my good friend here, Norm.

NORM RITCHIE: Thank you. We're very fortunate to have the DNS Glee Club in town. We're going to knock out a few skits for you. As we mentioned, DNS is a fairly dry topic. DNSSEC is even drier. There's a lot of terminology. But to really understand DNSSEC, you have to really understand how DNS works. So, we're going to act it out. We're going to do a DNS transaction now. This is typically how one might work.

We'll start with doing a bit of banking. I'm Joe User, and here we have various servers out there: Mr. ISP, the Root, Com, and Big Bank. So, let me start.

JOE USER: I'm going to do some banking. I've got to pay my bills, so I type into my computer, my browser, www.BigBank.com, and had it off to Mr. ISP.

MR. ISP: I'm sorry. I don't know that answer. Let me go try and find it for you. Hey, Root, do you know where www.BigBank.com is?

ROOT: Well, no, but I do know .com is. Why don't you try 1.1.1.1.?

MR. ISP: Hey, .com! I'm trying to find www.BigBank.com. Do you know where that lives?

COM: Hmmm . . . The website for BigBank.com, no, I don't know where that is, but I know where BigBank.com is. They're at 2.2.2.2.

MR. ISP: Hello, BigBank.com! Can you please tell me where www.BigBank.com is?

BIG BANK: Well, it turns out that I can! www.BigBank.com is at 2.2.2.3.

MR ISP: Awesome! Mr. User, here you go!

JOE USER: Thank you very much, Mr. ISP. Now my computer can go off to 2.2.2.3, which is Big Bank and I can pay all my bills. I'm happy to do that! I gotcha! Only two days late!

DAN YORK: That's how a DNS transaction works, typically. As you see me as a user, though, all I had to do is ask my ISP for the request, wait for him to do all his work with the servers, come back and get the answer.

Now, we'll do one with a man-in-the-middle attack. We'll demonstrate the same transaction, do some more banking, this time with a man-in-the-middle attack.

JOE USER: Okay, more bills to pay. More bills. Always bills! I'll sit down at my computer and type www.BigBank.com. Mr. ISP, can you help me out with the address, please?

MR. ISP: Sure, no worries. Happy to help. Root, can you please me www.BigBank.com is?

ROOT: I don't know where www.BigBank.com is, but I do know where .com is. Let me show you. It's here: 1.1.1.1.

MR. ISP: This is feeling awfully familiar. Dot-com, can you please tell me where www.BigBank.com is?

COM: I don't know where www.BigBank.com is, but I know where BigBank.com is. It's at 2.2.2.2.

MR. ISP: Hey, BigBank.com! Can you please tell me where www.BigBank.com is?

BIG BANK: Yes, I can. You can find it at 6.6.6.6.

MR. ISP: Cool! Thank you!

BIG BANK: Thank you!

MR. ISP: Here you go, Mr. User!

JOE USER: Oh, thank you very much, Mr. ISP. Now I can finally go to my bank, and if I've got any cash left, pay the bill I owe.

DAN YORK: Let's give them a round of applause for that one. [applause]

They're not done yet. They're going to get another chance to come back. What did they do there? That's the real process that DNS goes through all the time. All those things.

Now, we did leave out the one little detail that the ISP caches the answer. He doesn't go to the root every time. He's held on to that answer and goes through that process. I wondered if we should have just put the cap on there. It's coming on, okay.

The other piece is we talked about is the ISP. It's the resolver, which might be at your ISP. It might be at the edge of your enterprise network. It might be running on your own laptop for people who are doing that. But it's a DNS resolver somewhere. It might be one of the public DNS services that people use out there in some way. Something like that that's going on there.

The camera is blurry. They watched the skit and the camera didn't know how to auto-focus. Uh-oh, your chance for an Emmy might be . . .

So, let's talk a little bit about that. Again, we're going back as we talked about here, this process where Ogwina was trying to talk to Og on the other side and they go through this where they don't know where it is and now we want to look at how we add in the blue smoke and how we make that happen. This is where we're now going to introduce DNSSEC and pieces around that.

Again, looking back at this, the whole process comes in where we're trying to get to, in this case, www.BigBank.com and our attacker interjected and put this record in for somebody else. How do we prevent that from happening?

What DNSSEC does is it uses a digital signature to ensure that the information you are getting out of DNS is the same information that was put in there by the person who is putting it in there. The keys and signatures are all stored in DNS. There's new record, something called a DNS key record and a DS record, other pieces. There are some new records that come into play here. You can just look those up.

Part of what happens is when you go with DNSSEC when you go and send a query to get a record, to go get www.BigBank.com, you will also get the key that you need to check and the signatures you need to check. There are some other pieces that now come into play with this.

Part of this as well involves something that we call the chain of trust. In the picture that you see here in the skit when the folks come back up, the resolver (Warren in this case) knows where the root key is, knows how to get the key that begins that whole chain of trust and is able to know from one person to the next, from one resolver to the next, is able to trace it back up and know that nobody has messed with that key in the process. There's a whole way. It's all being done cryptographically, so if this all works, we have what we call global chain of trust.

The root of DNS was signed back in 2010, and along the way, various different top-level domains (TLDs) have been signed as well. Then second-level domains can be signed, and that's what we're encouraging people to do of course in here. That's how we make this chain of trust.

With that little interjection, I'm now going to call – sorry, we have one more stage. The chain of trust allows this, which is I can know through these use of DNSSEC signatures that this is the correct answer and I can refuse this one. Let's see that acted out. We'll bring our team back up here to show us this process from the DNSSEC side. Dr. Evil is going to try to get in the picture again. We'll see what happens.

JOE USER: Okay, more banking to do. More bill. ISP is still after me. Okay, so I'm going back to my laptop again. This time, though, the chain of trust has

been established. We have DNSSEC working for us now. I'll start off: www.BigBank.com. Mr. ISP, can you help me with the address, please?

MR. ISP: Sorry, still don't know it, but I'll go find out. Hey, Root, I'm looking for www.BigBank.com. Can you tell me where that is?

ROOT: Well, I can't tell you where www.BigBank.com is, but I do know where .com is. It's over here at 1.1.1.1. Let me just sign this for you.

MR. ISP: Looks good. Hey .com, I'm looking for www.BigBank.com. Can you please tell me where he is?

COM: I don't know where www.BigBank.com is, but I know where BigBank.com is. They're at 2.2.2.2.

MR. ISP: BigBank.com, I am looking for www.BigBank.com. Can you tell me where that is?

BIG BANK: Of course I can. You can find it at 6.6.6.6. Here it is.

MR. ISP: That signature doesn't look right.

UNKNOWN SPEAKER: Ha, ha, ha! It is at 2.2.2.3 and it is signed all the way!

MR. ISP: Let me have a closer look at that. Seems good! Here you go, Mr. User. I am sure this is the right answer.

JOE USER: Great! Thank you Mr. ISP. Now I can go to this address knowing it's not been tampered with along the way.

DAN YORK: Let's give these guys . . . [applause]. Thank you to Adam for stepping into and helping us on this.

That's the story of DNS and DNSSEC. That's how it works in this case. Subject to the fact that we're simplifying a bit, but that's the general idea of what's happening in here. Notice that now that Warren, as the ISP – this is Warren, by the way – he would now have that answer and he would be able to keep that for a certain period of time, so he would be able to keep passing that back to Norm as Norm did more transactions and more pieces around that.

This is how the flow of DNSSEC works. DNSSEC is all about ensuring that the information you get there . . . The reason I make that a particular point is it's not about encrypting the information. It's not about making it confidential. It doesn't do that. The focus is around ensuring you reach the right place.

Now, we showed it here for a web interaction, but DNSSEC is useful for any other kind of transmission across the Internet, any other kind of communication. In fact, one of the uses where it's being used a lot right now is in e-mail communication, coupled with something called DANE and some ways to provide records – we're actually providing TLS certificates – to provide encryption mechanisms.

So DNSSEC and DANE together can help provide a path towards more secure e-mail. It's being used in that environment. It's being used in the Jabber environment in the XMPP world for securing the communication between Jabber servers and other chat environments. So while we show it here as a web interaction because it's a simple use case for us all to understand, the ideas behind DNSSEC and other ways to use that are used for a lot of other things.

I see Paul standing at the back – Paul Watters – against the door there who has been very involved in using it for PGP and making open PGP records be accessible through DNS because you can put them into DNS, secure them with DNSSEC and now you know that you can receive an assurance that the key you're getting out of the DNS is, in fact, the one that somebody put in there. So it's it a very versatile system to use for all of that.

What I want to bring up next is my colleague, Russ Mundy, to talk a little bit more about why you might want to do this and what you can do with it. So I'll turn it over to Russ.

RUSS MUNDY: Thank you very much, Dan. Here's the clicker. One of the things that a lot of people wonder about is: why is DNS important anyway and why would you worry about somebody just getting in the middle for doing names work or stealing your name or giving a wrong answer?

Well, when people are doing that, it's not to get DNS or to confuse DNS. They're really doing it to get to applications, because essentially, every application that runs on the Internet today makes use of DNS.

So whether it's e-mail or Jabber or web, you're using DNS prior to actually making use of your application, although you never really see that as a user.

So, what can you do if you do that? Well, do a man-in-the-middle attack, just like we showed a little bit ago here. And if you can place yourself in the middle of application pieces talking to each other, you can do various things – steal passwords, collect e-mail.

One of the things that is sometimes done for these types of attacks is the traffic from the application is sent on to who's actually expecting it and they never even know that stuff was stolen in the middle.

One of the things that I observed a few years ago in just doing some general exploration about DNSSEC and DNS hijacks, there was actually a university that had as part of their coursework, the students were being included the requirement that they had to write a DNSSEC hijack piece of software.

Unfortunately, in the curriculum that I saw, there was nothing that talked about ethics or anything like that. They just said, "Oh, I'll go write some code to steal DNS out there."

The next thing, as Dan already said, the DNSSEC use of cryptographic mechanisms with DNS allows you to make sure from a user perspective that the answer you're getting is the information that was put into DNS in the first place. It doesn't give confidentiality to the information, but it does give authentication and integrity checks to the information.

The net slide will show how generally when you – there we go. When you first start a query, www.AB.org or BigBank.org, the first thing it does is go to your ISP, your local recursive resolver. It then, in turn, sends it off through the chain. The answer comes back and you can see there's actually – we're not getting answers. Am I pushing the wrong button? Here we go. There we go!

Eventually, after a bunch of packets flow around, the answer comes back and then you can make your application connection.

So a bunch of DNS stuff happens before the user's screen actually fills. In this case, the web application. But as a user, you don't see that. So what happens when you get a hijack here?

Here we go. I was pushing the button too quickly.

So if you hit off and there's no hijack, you get the same information back, whether you're using DNSSEC or not using DNSSEC. This is a web browser and a tailored website that's set up so that it shows what happens when you are making use of DNSSEC. In this case, there was no man in the middle, no attack. They give you exactly the same results.

Now we'll inject Dr. Evil Hacker. The query goes off the same way and Dr. Evil Hacker has said, "Oh! I know right now. I'm giving you the answer right away." And even though the DNS queries continue around

the network, the user already has an answer, so his computer was fooled. He goes to the wrong place. The answer eventually comes back to him, but the computer that the user asked the question from first has an answer, so it doesn't matter that the proper answer comes back later. He's going to act on the incorrect answer.

So instead of going where he's supposed to go, he goes where he's not supposed to go. But if you insert DNSSEC, then the local computer ignores the wrong answer and then takes the proper answer. So that's just a pictoral way of showing what happens with the hijack being prevented by DNSSEC.

So what happens in terms of what the user can see with a hijack? Same screen as before for the DNSSEC-enabled browser, but if you're not running a DNSSEC-enabled browser, you could have information changed, either the entire page or a part of the page. In fact, this particular illustration is one of a live hijack that we did at one workshop a while back. The screen is a little hard to read, but the hijacker in this case inserted information on the web page that was our friend Dr. Steve Crocker saying that DNSSEC won't solve world hunger. It will solve a lot of things, but it won't solve world hunger.

If you look at the one that's doing the DNSSEC check, there's no such entry there. So it can be an entire page replaced, it can be a part of a page replaced, it can be just – any DNS-based portion can be hijacked.

A lot of people think, "Oh, I go to a website. That's one DNS query, right?" No. Wrong. That was about five years ago, CNN.com, and it was about 60-some DNS queries [to fill a homepage]. Now it's about 150-200. So any one of those DNS queries can be hijacked, or multiples of

them. There's a lot of DNS that happens that's totally invisible to the user.

The biggest point of DNSSEC is to make sure that the actual DNS zone content is delivered – be able that the receiver can determine that they've received the right stuff. That's really what DNSSEC does is it makes sure the right stuff gets delivered.

So how many places does DNSSEC fit in this? Well, there's a little bit of DNSSEC in many places. So if you're running, say, a major operation that is heavily DNS-dependent, if you're a registrar or a registry operator, you have a lot of DNS and you probably have a lot of good staff that's very familiar with it, and you're probably going to be running it with your own organic resources. So you probably have a lot of DNS knowledgeable people.

But if you're a sort of in the middle kind of enterprise, you might be running it yourself or you might have outsourced it to somebody to run for you, or you might be doing a mix of it. Again, the DNS zone data is the important part, so whoever you are giving your DNS zone data to to handle, the DNSSEC is an aspect of that, because again, it is ensuring the correctness of a DNS zone data.

Don't let the crypto scare you. It's really just there to help make sure the zone data is correct. In this case, the simplest illustration is more data gets put into the zone that is the DNSSEC data, and the validating recursive resolver, if you remember Warren, went and got all of the things, checked the signature, and by golly, it just all worked and he got the right information back. So that's the simplest illustration of where DNSSEC fits into DNS.

So when you think about and look at your activities for doing DNSSEC, you need to look at your activities for doing DNS [south]. What you're doing with DNS often will dictate what makes the most sense for you to do with DNSSEC. That should be general guiding principle. I want to then turn it back to Dan as our MC to get into the question/answer part.

DAN YORK: Sounds great. Thank you, Russ. [applause]

So that's really what we had prepared in a way of prepared talks to talk about tonight. We want to cover a couple of little things before we get into things. You all should see a DNSSEC for Beginner's document floating around here. If you don't have copies, we have some more of those here.

On the backside, there's a whole series of links that are available to you that talk about where you can learn more about DNSSEC. Some of those are on the Internet Society site where I work on the Deploy 360 program. We have some on the DNSSEC Tools project that has a number of different tools that are out there. The dnssecdeployment.org, some other different pieces that you can use on here, including a program called OpenDNSSEC, which is used in the signing side of things. There's a version of Mozilla called Bloodhound, which is a browser that is used for secure DNSSEC browsing, you could say. It has the validator build into the product or into the application, so that you could use it in the case that Russ showed where you have all those hundreds of different queries. It would validate all of those.

There are also some plugins available for some of the other browsers that let you go and at least see the DNSSEC status and some of the places that are out there.

This is some information you can get back on here. Some tools from VeriSign Labs. There's a bit about the history of DNSSEC and other pieces that are there.

The other thing I'll just mention right now before we go into questions is if you're interested in more, on Wednesday, we have our Deep Dive DNSSEC technical workshop that happens in this room starting at 8:30 and going until about 2:45, and we have a long agenda that's published on the site that has a lot of different things ranging from DNSSEC activities here in Asia-Pacific to discussions of tools, discussions of monitoring solutions and what should be a very interesting panel around DNSSEC and DNS operators and how that all works. If you're interested, come to that on Wednesday at 8:30.

With that, I'd like to open it up for any questions. We've got a whole pool of people here who can answer it, so this is your moment to ask them. I see a gentleman in the back. Let me come to you with the microphone because we do have people who are remote. Do we have another mic? We got another mic, awesome.

UNKNOWN SPEAKER: First of all, thank you for that great explanation and great representation of such a complex thing. Every time I bring up DNSSEC with more knowledgeable people, always after a few minutes

[inaudible] enumeration attacks. Could you please elaborate a bit on that? Is that a real thing or is that something that could be [inaudible]?

DAN YORK: Sure. The question was about enumerating the zone. Basically, [walking] the zone and being able to see it. In the initial way that DNSSEC was first created, there is an attack that you could use to go back and walk the zone, basically, to know what were all the different other elements that were there.

Now, that happened when people implement something called NSEC. There's also now something called NSEC3, which prevents that by adding a little bit of a hash inside there to go and create that.

The other piece about that is oftentimes, for a lot of the public zones, the fact that you can walk the zone and be able to find all the records doesn't really matter in many cases, especially on public sites that you're looking at. It's not really as much of a concern. I see Russ wants to weigh in.

RUSS MUNDY: Yeah. Early on in the DNSSEC design, the general expectation and philosophy is that the total content of DNS was not of – releasing the information was no concern because anybody could get to it anyway.

That turned out to be a faulty assumption. For certain zones, that's still a very, very valid assumption. For instance, the root zone. The content of the root zone is completely public. Lots of people get it. Lots of people look at it. Some TLDs are considered public. Some are not. That

was the reason that NSEC3 was created is to accommodate those that needed to actually protect the full content of their zone. Individual answers are fine, but they didn't want to release the full content of the zone.

DAN YORK: So the answer is that they can look at implementing what's called NSEC3, and there's a lot of documentation out there and different ways.

Somebody else had the other microphone I think. Or no? Okay. This gentleman over here. I saw you, too. Go ahead with your microphone there.

UNKNOWN SPEAKER: Actually, this is my first-ever session relating to DNSSEC, so I'm sorry if the question comes across elementary. In your example, the consumer actually new which website they had to access, but most of the traffic which gets directed to any website is via a search engine. So how does that work in terms of the search results get?

And sometimes we notice that at a different location, like if I was to search for some keywords in Pakistan, my top 100 searches would be very different from if I were to do it in Singapore. How does that all relate? Thank you.

DAN YORK: Sure. Likewise, in this world of personalized search engines, even if you and I are here, my results are going to be different from yours.

The fact, though, is when you get those results, you're clicking on that link and even though you may not have entered in www.BigBank.com, the link you're clicking on is going to that site. So the search result is really just getting you that information, and when you're clicking on those links, that's when you're initiating this process and you're going and contacting your DNS resolver and saying, "I want to go to this site." That's really what's happening there.

Then it's ensuring that you're getting the right answer back, that the owners of BigBank.com, they put in www = this address. So you're getting an assurance that that is the address that they put in there. So it's going back to there.

Please, feel free. We had a lot of people in here who had no exposure to DNSSEC, so this is your moment to give us questions that you think might be really, really basic, throw them out here. We've got people. Wes wants to say I think in response to you.

WES HARDAKER: Yeah. One other really quick point is that using a search engine gives you actually two different places to possibly poison them. The one is that you can redirect them to your search engine instead of the one that they were originally trained to go to. And then two, as Dan said, the link from the search engine needs to be looked up, too. So there's actually multiple points of attacks by always going to search engines.

DAN YORK: And if you were to go to searchengine.whatever – pick your search engine you wish to use – DNSSEC, if that domain was signed for the

search engine, the domain that you get back, you could be sure you're reaching the correct address for the search engine and it hadn't been redirected by an attacker in some way.

I saw the gentleman over here.

UNKNOWN SPEAKER: Actually, my background is [inaudible] fellowship from [inaudible] organization of [Sudan]. My background is [inaudible], and I just come by the [inaudible] as a hobby, but still DNSSEC is very – it is not clear to me because I am working just on the browsers. I am not going deep. Sometimes, if I'm not mistaken, I get a message that this page is encrypted, do you want to continue or to quit? I didn't get the right answer. Sometimes I will continue. Sometimes I go quit. So which is the best way to tackle that problem?

The other thing, I had my website hijacked three times and most of the time, all my data are lost because I am not familiar with that. So, in the future, how to tackle that? Thank you.

DAN YORK: So you've got a couple of different questions in there. The first one, as far as when you get that error that says this may not be protected or may be signed at the wrong certificate, that's typically what's happening with your TLS or what commonly is known as SSL, but it's really the transport layer security. It's a certificate that's being used to encrypt the connection to that.

One of the things that's interesting is the experience we've had as users with those warnings is that very often people just click through those warnings. You get the thing that pops up: "Hey, this is the wrong certificate. What do you want to do?" Well, most people want to get to the site, so very often they just go and they click through that and they go on to the website, at which point they may now be talking to the completely wrong server in some instance.

One of the interesting things with DNSSEC is that it doesn't – the implementation. It just gives you back a failure to get to the website or get to the IP address. You can't get there at all. If you want to dive in there, it's called a SERVFAIL is the error in DNS. It basically says, "Sorry, that address doesn't exist."

The challenge that you get into is this can create an interesting user experience. The folks from Comcast gave us the case study a couple of DNSSEC workshops ago where they talked about the case when they rolled out DNSSEC validation to their large network in North America.

What they did was they started to anytime a domain didn't validate correctly, then the end users would not get the website. Well, what happened was the folks at NASA – the US space agency, NASA.gov – what they did was they messed up. They wrote a whole white paper about this, by the way. This is all public stuff. They didn't handle the key rollover of a new key correctly.

So all of a sudden NASA.gov stopped validating. It had a bad error, so people who were going to NASA.gov all of a sudden couldn't get there Comcast network.

Now, of course, this being the age of social media, people whipped out their phones, looked at this, said, "Oh, I can get to NASA.gov on my mobile phone," because they're not doing DNSSEC validation. "It must be conspiracy on Comcast's part. It became this big thing on Twitter and everything else.

The reality was they were blocking people from getting to what was a bogus site, in their perspective.

So with DNSSEC, the failure that we're seeing is not something that you have to click through like that. It's typically more you just won't get to the website at all. You won't see it. It will never come back to you. That's typically the implementation that we see with that. So it's providing you with a protection layer on that.

As far as how to protect your site from being hacked and things on that line, I think we could have a larger discussion there than what we could really talk about with DNSSEC here.

Yes? A question back in the back?

UNKNOWN SPEAKER: Yeah. My small or less like an observation. A similar incident happened in Ghana. That was last month. My name [inaudible] from Ghana. I work with [inaudible] Agency.

What happened was about seven websites of government agencies were attacked by hackers from I think Algeria and then Morocco. Then what happened was our president made a statement that Africa

[inaudible] the Western African region has to combine forces and [inaudible] people.

So then the following people, our president happened to travel to Germany. I think the German president invited him. Then they were talking about energy issues or [inaudible]. I think within that same period the hackers attacked those seven website. The information they left was they are the [inaudible]. You made a careless statement and then we are going to kind of [inaudible] you down. That was their statement, the information they left on the website.

So we did an investigation, and in the investigation it came out was the attack was at the [application] level. What we realized was most of those websites were using Joomla, and then there were notes updating the [inaudible] and those things. [inaudible] log in and other things were not protected. It was open.

So they used that, and then they sent a broadcast message. It was a robot. The robot happened to enter in, brought that information and they were able to [inaudible] enter into the system. Through that, they [inaudible] information on the system and they replaced the index file with their own file with the information.

That was that incident that happened. So we had to bring all of those technical [inaudible] from within those [inaudible], and then we gave them the training that this is what goes on. Especially with the forms that you have within your website, you have to at least put an image CAPTCHA on it, so that when those robots come, it would be too difficult [inaudible] CAPTCHA for them to be able to enter into the system. So he's saying they attack his website three times, so I was

thinking maybe the loophole was it was not updating security and he was also . . .

DAN YORK: Right. That's a great illustration of the kind of attacks we're seeing out there today, and it's a very common one. A content management system isn't being updated or just the website isn't being updated in some way.

This is an illustration, though, too, that the attacks we see are at a bunch of different levels. DNSSEC provides a way to protect against one layer of the attack. It ensures that people are in fact getting to your web server – and to the correct web server. It doesn't do anything to protect that web server, per se, from being attacked in the manner that you outlined. It's really just one of the different elements if we think about it in a number of layers of defense against the attackers that are out there.

So a signed record would prevent people from going to another site, if it had been redirected. You outlined the attack against the actual server that happened. Those are the terrible kinds of attacks that are happening all the time.

There is another one, though. Another attack that we're seeing out there in the wild where people are doing exactly what you talked about here. They're redirecting the DNS so people don't even ever reach your web server. They're going to another one. What? Malaysia Air about two, three weeks ago was having their website being redirected somewhere else.

There's an attack that's been ongoing for some period of time that the researchers at Carnegie Mellon, the cert cc at Carnegie Mellon, indicated that their research showed that people were redirecting e-mail through unknown servers by providing the MX records, which are the mail exchange records that you get back when you're delivering e-mail – somebody out there, and the cert cc researchers couldn't identify who, was redirecting e-mail to some other servers and it was ultimately being delivered, but in the meantime somebody out there was siphoning it off and looking at the e-mail in some way.

These kind of attacks are where DNSSEC can help, because if the mail server were to be able to check the record and know, "Yes, this is the right place for me to go," then it could prevent that kind of attack.

I saw a gentleman over here.

UNKNOWN SPEAKER: So we had the question about a search engine. I was wondering, are there any special requirements for the search engine providers – Google and the rest of them – because we, as a community, are trusting them to tell us that this address we're sending you to is the correct address?

DAN YORK: Oh, okay. Julie is asking, too, if you could state your names please for the folks who are remote as well. Name and affiliation. I'm laughing because Warren is slinking down over there, and Warren is from Google. Do you want to say anything about that?

WARREN KUMARI: Well, there's not actually a huge amount that the search engine can do at that point. They're just giving you a URL, the name of the next site to go to. So what needs to happen is for those sites to have some sort of DNSSEC protection.

If you encourage people to sign their zones, then whichever search engine will give you the URL and then you'll be able to know you're going to the correct one. Once a search engine gives you an answer, though, it's completely out of the picture.

DAN YORK: But you have a larger – there's a larger question of search engine ranking and how if you do a search under your organization's name and what comes up. That's a larger question, really, in the space of search engine rankings, etc. I saw somebody – yes? I see a couple questions here. Okay?

ROHANA PALLIYAGURU: My name is Rohana. I am a first-time fellow from Sri Lanka. What I need to know is, as a client, how do we know that the particular domain is using DNSSEC?

DAN YORK: How do you know if it's using DNSSEC? Okay, that's a good question. A couple of different ways. One is if you have a validating DNSSEC resolver, it will query that and it will get back – there's a certain bid it can get back to say that there is a signature there. You can also get the records for it. There's what's called RRSIG records. There's additional

records that would come back to you that would know there are signatures associated with this domain.

When a query goes out there, you won't just get back the record for the website or whatever else. You'll get back additional records as well. There are some ways that can happen on there.

ROHANA PALLIYAGURU: Is there indication in the browser?

DAN YORK: No. The question was: is there an indication in the browser? There's not, and that's one of those questions that we get into about the user experience and the value of the lock.

Now, if you want to have it in your browser, you can use one of the plugins that's mentioned in the back in here. The folks at CZNIC Labs from the Czech Republic, they have made up some nice browser plugins that do provide you that kind of visual clue for the main URL that you go to that's out there.

As Russ mentioned, when you go to a website, it's querying a lot of things, but it would at least check the main URL that's there and you could see that, if you're interested. Beyond that, the user experience is [inaudible]. I saw you and somebody else in the back there. Go ahead. Yes?

[LUCY]: Hello. My name is Lucy from New Zealand. We touched a bit on the signatures and it makes sense to me, but I was wondering if you could explain a little bit more how the keys work. You mentioned that [inaudible] that all went kaput. It seems to me that whenever there's a chink in the chain, someone will eventually find a way of [inaudible] it. So how are the keys protected and work to make sure that no one manages to step in there?

DAN YORK: Sure. When you sign a zone, as you have there, when you sign that, what happens is it's a standard public-private key thing that you generate a key that's used to sign the zone. Then there's a public key that's shared out publicly that people can use to validate the zone. If you've ever done anything with public key cryptography, it's that kind of mechanism. So there is a good protection that needs to be done around that private key, so that nobody can go and steal that and sign on your behalf.

There are documents called DPSs (DNSSEC Protection Policy Statements) that outline the steps that some of the TLDs go through that really makes sure that the key is protected. In fact, every three months, ICANN goes through what's called a key ceremony for the root key where they generate a new signing key, and they have a whole long scripted program that you can go and watch the live video of to see actually what the steps they go through to ensure that they have a very high level of accountability around that process.

Now, that's beyond the scope of – for my own domains, in a couple of cases, my involvement is I use a DNS hosting provider where I go in and

I check a box that says "Enable DNSSEC" and – boom – it's done. I'm trusting them to manage my key and to do so, and they've done so. I'm happy with the security that they have.

In another case, I'm using software running on my own server, and when I've set it up, I had it generate the keys, so the keys are existing on my server and they have a length of time. There's some best practices around how you do that and there's differences with zone signing keys and key signing keys and stuff that gets a little bit more involved than we care about today. But the net of it is that typically you would sign a key for a year for a top level. I'm getting off into the weeds.

Let me just leave it there and say there's a time period that you'd sign that for, and then you'd roll that at the end of that period, typically, some way.

I saw a woman right there in red right behind there for a second. Then I'll get back to you there.

CATHERINE NIWAGABA: Thank you so much. My name is Catherine. I'm a fellow from Uganda. You've touched a few, in your response right now you've talked about a few what I was intending to ask.

One was some specific technical requirements for a business or a company to implement DNSSEC. And also, the user experience. You said there is not so much on the user experience to discuss now, but just wondering, if you're getting heat, [inaudible] DNSSEC protecting you. Is there a way you can know, like a log or any tools you can use, for instance, to say DNSSEC has protected you from so many attacks and

then maybe you can satisfy that DNSSEC is actually a good investment for you?

DAN YORK: Sure. Let me answer your first thing about what can a business can do. There's two parts to DNSSEC, and it's important to think about it as two parts. One is what we talked about here, the signing of your domain or domains. The second part is the checking of signatures, the validation, doing what Warren did as the ISP. All of you today could go home and turn on validation in the DNS resolver that you're using. If you're using BIND or UNBOUND or Microsoft Windows Server or any of these, it's very often a single configuration line in a configuration file that will turn that on and start you on the path of checking signature.

That's something that all of us can do. I've done it in my home router in my own system. I'm doing that in my home network. You can turn that on and enable DNSSEC validation in a DNS resolver. That's one side of it. It's a checking of signatures.

Now, separately, you can go through the process of signing your domain or domains, and now that may be a little bit more involved, because you need to involve whoever is managing that zone for you, managing the domain, which might be your registrar if they're also being a DNS operator. It might you or your IT team if they're the ones who are managing it. It might be another DNS operator who's doing it.

So that may be a little bit more – the signing of the domains might be a little bit more involved, but the turning on the validation is something that any business can do and start to get the protection around that.

As far as the tools around the – is that what you wanted to . . . ? No, I was going to say as far as – there are a number of tools . . . Wes, why don't you answer that? I'm talking too much. Wes Hardaker, who has been involved in a lot of DNSSEC tools.

WES HARDAKER: I think the one of the primary points of your question is how would you notice when there's errors. All of the name servers that will do validation for you do spit out errors in the log file, but you often have to go look for them. From the user's point of view, the web browser is just going to get this note saying, "I failed to look up the name," if they're using a validating resolver near them and you won't be able to tell exactly why. You just won't be able to get there.

In the same way that if you mistype a name into your browser, if you type Google like Gooogle – which there probably is a Google with three Os. But 15 Os. If you typed Google with 15 Os, it's not going to be there. You'll get that exact same error.

What will happen is if you actually type Google correctly and you're going through a validating browser and it gets an incorrect answer and the little signature was not on the back of the paper like you saw earlier today, you just failed to get there and your browser would say, "I don't know how to get to Google.com," because it doesn't. It was returned in error.

What happens is if somebody goes to look through the log file, they'll find out that, yes, there was actually a failure to get the correct answer and it had to give up.

DAN YORK: Warren is probably checking to see if Google with 15 Os actually works. Okay, I saw this gentlemen, and then I see two folks over here. And I see two more over here. Okay. All right, yes?

[MOHAN BUTRA]: Hi. This is [Mohan Butra] from National Internet Exchange of India. My question is if a user knows the IP address directly of the website and types it in the web browser, does DNSSEC come into the picture in this case?

DAN YORK: No, because they're just going directly to that website, so they're not using DNS at all in that case.

[MOHAN BUTRA]: Okay. So there's also a [inaudible] record I think in the DNS. So [inaudible] DNS resolution.

DAN YORK: So if they use – PTR records, reverse lookups. In some of the – we need Roy here. He was working with the—

Some of the RIRs have been signing the reverse lookup zones, so you could – a PTR record that would resolve that would come back with the signed information. That would be a case where you would see that more in your log files, for instance, or some mechanism like that. Is that where you're going with that?

[MOHAN BUTRA]: I'm not very sure about – but another question, small question. There is a thing called Dynamic DNS as well. Dynamic DNS?

DAN YORK: Yes.

[MOHAN BUTRA]: So does it have anything to do with DNSSEC?

DAN YORK: Dynamic DNS is really a service typically where you're registering a server. I actually think I saw this . . . Okay, well, we'll get to you. With Dynamic DNS, what's typically happening is you're updating the service with your new IP address.

Now, what gets interesting is every time that a DNS zone changes, it has to be resigned. So if you're updating that, then the Dynamic DNS service needs to resign the zone.

There are services that do that kind of thing where they do the online signing and they can just do it right away and very quickly and do all that kind of thing. You would need to find out if that Dynamic DNS service would support that kind of in-line signing or online signing that was there.

I could see that being more resource intensive for some of those services. I'm not quite sure about what dynamic DNS services are

supporting DNSSEC signing at this point. I don't know. That would be a good question to ask, though.

Okay, over here.

ORATILE SLAVE: Good evening. My name is Oratile, a fellow from Botswana. As a legal person sitting in the Technical Advisory Committee for the .BW domain name ccTLD, I was wondering what role I can play in the deployment of DNSSEC for .BW because [inaudible] is in the process of deploying the same for the .BW ccTLD.

DAN YORK: Excellent. We'd love to be able to – if you'll look at the DNSSEC deployment maps, it would be excellent to add Botswana in there. We need some more coverage down in there. The first step to being able to sign a domain is that you need your top-level domain – your TLD needs to be able to be signed. So you have to have that. That's excellent that you're doing.

So as a legal advisor or that role, what would be the role, gentlemen over here? What would you suggest for her as far as what she could do to help with that?

I would say one of the things would be to encourage, for instance, government departments to sign their domains and anyone else who's part of that ccTLD. That would by certainly one aspect.

For instance, the US government is an example, the .GOV domain. They had a mandate go out a number of years back that said all government

agencies should be DNSSEC signed by a certain date, etc., and they're up to now about 87% of all .GOV domains have been signed.

The government of Puerto Rico is not here, but the gentleman is here at ICANN. I just spoke to him at the last session. They've done a lot of work. In their case, the .PR registry works with the government and they've set up a system where they're actually automatically signing all the government domains. So they've set up a system so that all those .GOV, .PR domains are being automatically signed by the registrar.

So it may be something that you could help with in that mechanism. Any other suggestions for her? Jacques, anything from Canada? Do you want to – no? That would probably be the starting place would be to help in those kind of ways.

The other piece, too, that we're trying to – and this is something that might be of interest, too, is to talk to people around the fact that you are helping secure your identity and ensure that people are being able to get to the correct sites.

That's certainly been something, when people talk about security about protecting their online identity, their online brand and pieces like that, they're using this as a way to help ensure they can get there.

Yes, Adam?

ADAM LEACH: Adam Leach from Nominet. We look after the .UK zone and we have a free DNSSEC signing service to people that want to sign their domains.

DAN YORK: He said in .UK they put a free signing service to help people sign their domains, so they've implemented some pieces around there. I see over here?

UNKNOWN SPEAKER: Thank you. [inaudible], .UA.

DAN YORK: I'm sorry, what?

UNKNOWN SPEAKER: .UA.

DAN YORK: .UA, okay.

UNKNOWN SPEAKER: Yes. Do you have statistics how much cybercrimes were prevented by using DNSSEC is the first question. The second, does cybercrime have their own solution against DNSSEC might be more complicated hijacking attack on something like this.

DAN YORK: I don't have any statistics, nor am I aware of any, that would talk about how much cybercrime has been prevented by DNSSEC, because partly, again, DNSSEC is one layer in a much more comprehensive suite of defense mechanisms that there.

We know it has prevented some hijackings and some spaces that are out there, but we can't necessarily quantify it. There are a number of statistics sites that talk about the amount of implementation, the amount of pieces that are out there, the amount of validation going on. That would be great research. I'd love it if somebody would dive into that, actually.

I see a number of other questions.

ZAKIR SYED: This is Zakir Syed from Pakistan, an ICANN fellow for the record. I have a quick question. Is there an alternative to DNSSEC? Is it possible that we use some application layer firewalls or some sort of software to be able to overcome the issues that might arise because of not using DNSSEC?

And I have a follow-up quick question. Is this something similar to IPSec? We heard a lot about traditionally known as IPsec. Is IPSec something for the numbers at the same levels and DNSSEC for the names at the same level?

DAN YORK: All right. What was your first question?

ZAKIR SYED: An alternative to DNSSEC. Can we use some alternatives like—

DAN YORK: Oh, yes, okay. The answer is DNSSEC solves this particular question around how do you ensure that you get the information back from DNS.

To that solution, that DNSSEC is really the answer that we have to protect the integrity of the information coming out of DNS. That's really the solution that we have today for that.

Now, when you start getting into application layer questions, when you look at how to use, for instance, TLS to provide an encrypted connection between your web browser and your web server, there are other mechanisms that are being used to help ensure you're using the correct TLS certificate, for instance. Using the correct encryption certificate that's there.

One of those that we talk about in the DNSSEC [inaudible] is something called DANE, which is putting a fingerprint or the entire TLS certificate into DNS signing and DNSSEC, etc., and it provides a mechanism to know that you're using the correct encryption certificate for that.

There are also other mechanisms. There's certificate pinning in browsers. There's other different pieces that get into play that can help with that kind of mechanism at the application layer. But at the DNS layer, DNSSEC is really the solution that we have at the moment to know that you're getting the right information out there.

On the IPSec question, I'm going to turn it over to Mr. Paul Watters here, who is Mr. IPSec and other things.

PAUL WATTERS: Hi. The traditional IPSec deployment is a VPN deployment where both parties know each other. So I have a VPN client. I have configured a certificate and a server for my VPN server on the other side and I connect to a known party. It works really well if you're connecting from

home to your office, like your office has given you the information you need to connect to them securely, verify their identity and then encrypt everything.

The problem is I cannot configure that for millions and millions of zones to do the encryption and everything, at least until now. So one of the things we're working on is actually a DNS record called IPSec key, where you can put the IPSec key, the public key, into DNS. And because of DNSSEC, you can now prove it's the right key and now we can start rolling out encryption, like VPNs, to everyone without any pre-configuration.

So I have a few servers running some of this experimental code. If you go to [inaudible].ca, there will be an IPSec server running and the key is published in DNS.

We're hoping to get more encryption automatically going without any manual work, because it will never scale if we have to do this all manually.

ZAKIR SYED: Just a quick follow-up question. Since we don't have an alternative to DNSSEC, is there any – that means every TLD - actually, every registry – requires DNSSEC to be deployed, right? So do we have some policy compliance or some enforcement from ICANN or maybe from IANA that every registry has actually to . . . ?

DAN YORK: Well, yes. Let me also just be clear on one thing to what Paul said. So IPsec and DNSSEC do two very different things, too. DNSSEC, again, is ensuring the information you get out of DNS is what was put in. So it's integrity checking. IPSec is actual encryption, so it's encrypting the tunnel – it's encrypting a connection that you have. So one is protecting the confidentiality of your connection and the other one is ensuring the integrity, making sure it's there.

So they are two different pieces that work together complementary in the way that Paul mentioned where the two can be used together to ensure that you have a really [tight thing].

So, yes, TLDs need to be signed in order for the sub-domains underneath that, the second-level domains, to be able to be signed.

On Wednesday morning, I'll show you a set of slides that have maps and counts about what we have. We're now up to over 77% of all top-level domains are now signed with DNSSEC.

Now, largely because that number is so high, because in the new gTLD program, all new gTLDs must support DNSSEC from the beginning and must be signed at the top level.

So as we've had these 500+ new gTLDs come into the overall network, they're all coming in signed. Anybody who registers a second-level domain underneath a TLD has the capability that they could sign it and have it linked into the global chain of trust. Again, this means that TLDs are signed. The second-level domains don't have to be, but they have the option.

A number of the ccTLDs are the ones that are in that remaining 23% or so that are not signed are a number of the ccTLDs, and some of the programs that are out there to help with that include programs from ICANN and their DNSSEC team that are going around working with a number of the different ccTLDs to help them get signed and to work with that.

Hearing the woman from Botswana who was talking about their work there, there's a number of other domains that are turning on. We'll talk on Wednesday morning. Australia actually just recently signed their domain as well or published it from the root on down. So there are some [inaudible] happening there. Hopefully that helps a little bit.

Compliance-wise, though, it depends upon whether you're a new gTLD where it's required. ccTLDs it's certainly encouraged and we're hoping that they will go and do that. Nobody is telling you that you have to, right? No. Jacques is with .CA, so I can say that.

A gentleman in the back has been, I see – somebody needs to track it go ahead.

[STEPHAN]: Hi, this is Stephan from Malaysia. I log into a wireless router here, which I'm trusting the [inaudible] of the current network. So I saw there is a primary DNS, a secondary DNS all assigned to me by the router. So what's stopping some guy from just hacking it, change the router, malicious, point me to the bogus DNS server and then I'm done. Is there any way to prevent that?

DAN YORK: Not really. Okay. Paul, Russ, and Wes want to jump in here.

RUSS MUNDY: So I think the easiest way to counter such an attack is to simply run a recursive server on your own machine that does DNSSEC validation, then you know you are using DNSSEC and that it is being properly validated because you're in control of it.

[STEPHAN]: But I won't be having the full list in my computer, right, if that's the case? How do I do a recursive DNS lookup on my [notebook]?

RUSS MUNDY: You can put a recursive name server on your machine and it will run just fine there. Many of us do that.

UNKNOWN SPEAKER: In fact, many use something called DNSSEC Trigger in combination with a resolver on our laptops and some people do it on their phones. What you're using is you're running a validator on your own machine, and when the network gives you a DNS server, you're only using that to get the information from outside the world, if you can use it.

The good thing about that is you're still using the cache. We're all using the cache of the name servers assigned by the ICANN network here, even though we're all doing our own validation. So we're not trusting it, because DNSSEC data is signed. We can ask any stranger for it because we know when it's been tampered with. So we just as these insecure

servers that ICANN provides us here on the wireless, which we don't trust, and we run them through our own validation to make sure that they got their [inaudible] or not.

Now, there are some trick when you're at hotels or at coffee shops where they make you believe a few DNS lies for you to go to accept the terms and conditions or put in your credit card number. So sometimes you have to believe the lies before you can get on the network. That is actually the biggest problem for us.

We have to come up with all these innovative ways where we allow these lies to happen for a little bit until we don't them anymore, and we have to make sure that they don't contaminate our existing DNS cache, so that they don't have this 15 seconds between you clicking OK on the webpage to [inaudible] to you. That's actually fairly hard to do, but it's getting there.

DAN YORK: Yeah. That's something called DNSSEC-Trigger. That's one of the things that's out there. This is a challenge that we have in terms of even if you get the answer back, what's the zone? Could somebody go and inject an answer in there? Could Dr. Evil swoop in and do something like that?

Because you'll see people, for instance, who will use Google's public DNS – the 8.8.8.8 – or the IPv6 equivalent and it does DNSSEC validation. So all the people out there who are using Google's public DNS are getting DNSSEC validated answers, which is huge. It's an enormous number of people that use those services in doing that.

The challenge is that there is still space. You're sending your query out to Google's public servers waiting for the response to come back in there. There is space in there for the Dr. Evil to swoop in and give you an answer back faster than Google's public DNS answers could get to you, which is why ideally you're running that DNS resolver as close to you as possible, even down to running it on your own machine or on the edge of your network.

The closer you can run it to where your applications are, or even in the application itself – like the Bloodhound web browser I was talking about that runs it in the application itself or in the mail servers that run it right inside the application – then the zone for the attacker to jump in is much smaller.

Wes, you want to say something. I can tell because you're standing there.

WES HARDAKER: Actually, the mail server in question requires a local resolver, but anyway. The important thing is I think if you look 20 years down the line that you'll find a lot of our applications will do validation right in the application or they'll have a trusted link to their local resolver where they can get error codes.

It actually brings up a really valid point going back to the woman in the back who asked about user interface and how can you detect things. Well, right now you just get an error. We have a validating libraries and there's a few other validating libraries out there that you can put directly in your application, and then all of a sudden, you get all of the

diagnostics for exactly why an error happened, what happened because of it, and you can actually present that to the user more than just saying it failed. You can tell them why it failed.

So I think if we look 20 years down the line, we're slowly pushing validation closer and closer into every application DNSSEC or other security technology. It's all happening in the end.

UNKNOWN SPEAKER: May I?

DAN YORK: Okay, yeah.

UNKNOWN SPEAKER: I use [TOR] sometimes. Will that have any protection or any way – no, not really, is it?

WES HARDAKER: [TOR] doesn't provide authentication, so you're still going to have to do the same security protocols down to your laptop no matter what tunneling mechanism you're using. You could tunnel all your traffic over SSH, over VPNs, over [TOR]. It really doesn't matter. At some point, you have to make the decision on your laptop: is what I asked secure or not? Then you have to have all the pieces to make that decision, but you still need it on your laptop in that case.

DAN YORK: I saw a gentleman in the back has been up there.

FAWAZ SALEEM BOKHARI: This is Fawaz from Pakistan. I am a first-time ICANN fellow. My question is what happens if the website does not exist, there is no record in the DNS and then the response that will come back from the DNS, how is it going to be signed? Maybe someone can just say that this website is not registered. So what is the procedure for that?

DAN YORK: Sure. So part of DNSSEC is this idea that you assert that records do exist and are the correct ones coming out of there. There's also the assertion that they don't exist, which is this whole thing we talked about with NSEC and NSEC3, the way these records are there, that if the website didn't exist, you would get back a signed assertion from the resolver that that site really doesn't exist. You would know that.

If an attacker – all right, let me walk through this quickly. If an attacker were to remove an entry in some way, but they didn't have access to the signatures, you'd have the same kind of thing. They'd be coming back saying, "No, it doesn't exist." But wait, you're not signed. Get out of here. And then it would wait to get a query back that would say, "Here is the real record." So DNSSEC does provide protection against that through the signatures to ensure that it is there.

UNKNOWN SPEAKER: [inaudible]

DAN YORK: What would happen is you're querying for research.bigbank.com, whatever else, something like that, and it goes out there. It would come back telling you that that site does not exist and giving you a signature to say, "I am telling you and I'm signing that it doesn't exist." Russ, do you want to . . . ?

RUSS MUNDY: Let me explain it this way. The normal DNS response for something like that, that it doesn't exist, is you don't really know if it doesn't exist. DNS comes back and says, "I couldn't find it. You might try again in a little bit. Maybe it will work. Maybe it won't." DNSSEC says, "No. It's not there." That's an increased functionality that you get with DNSSEC. It definitively gives you an answer about something that does not exist.

DAN YORK: Right, that [inaudible]. You know that. I saw some other questions floating around. This gentleman, yes?

UNKNOWN SPEAKER: No, actually, I had the same question [inaudible].

DAN YORK: Oh, okay.

UNKNOWN SPEAKER: I might want to add something. You said that a solution would be to have your own recursive DNS, but this cannot be a global solution for all

users unless the DNS server is included a new browser in the operating system. Is there any effort towards this kind of solution?

DAN YORK: I think there's a couple answer, and one is that we're starting to see DNSSEC validation being brought down into operating systems. For instance, Fedora. The next version of Fedora that's coming out will have DNSSEC validation enabled by default. Now, granted, that means you might run Fedora on your laptop or something, but if you did, you would then have DNSSEC validation happening there on your laptop.

Now, it might be that you might run that on the server gateway that runs on the edge of your network, so again you'll have that there. I think what we've seen in the path of validation has been it started out doing DNSSEC validation at the ISP networks, and things like Google's public DNS and other places like that. We're slowly moving down and down and down.

Wes mentioned that applications now have this library that they can use. There's [on the] DNSSEC Tools. There's another one called the getDNS API that's been developed by VeriSign Labs an NLnet Labs that provides again a DNSSEC validation that can be built right into applications.

So more and more of these tools are coming out, so that people can do that in the application. We see that. Some of the largest mail servers are now having that DNSSEC validation happening as part of that. So we're seeing it pushed down in that space, and that's the evolution that I think we'll continue to have as more and more of this rolls out.

Russ?

RUSS MUNDY: Yeah. I was being a little sloppy in my terminology, which I hate to do, because I did say a recursive resolver and that's right, but that's where the recursive resolver is actually doing validation and the most important aspect is actually doing the validation whether it's on the laptop or in an application that's running on the laptop, and there's more clever ways that are being looked at and put out there that allows – you may be doing the recursive resolver on your [own] machine, but you may be able to take advantage of recursives around you, because the thing about DNSSEC that's very important – it was an early design decision made – is that it's the data itself that you're validating.

So as long as you get the set of data that you need, whether you're doing the recursive lookup or you're using some other machine, you can do the validation. And that's the important thing is to do the validation as close to the actual end use as possible.

DAN YORK: I've got time for a couple more questions. I see this gentleman and this gentleman. Anyone else in the back? Okay.

[SAHID]: Hi, my name is [Sahid] representing the civil society from [inaudible]. You said that 77% of the TLDs are using DNSSEC. Can you tell me who are the major parties in the 23% of the remaining and how long will it

take to raise the number? And what sort of collaboration are you expecting or requiring from them?

DAN YORK: So, to be clear, the stages for a top-level domain or for people to use DNSSEC is that the TLD has to be signed. Then, at that point, second-level domains can be signed and they'll all work in this chain of trust that goes on there.

If I had my slides for Wednesday, I could show you – or Julie, actually, could we pull up? You know what, we're running out of time to do that, but I could show you. Actually, if you look on the back here, I think there's a link for – the DNSSEC deployment maps are on here. You can see a picture there of where the ccTLDs are that still are not – they're in that 23%.

I mean, a lot of them are in Africa, South America, some parts of Asia in different places that are there. What we're doing to help that is a couple different activities.

One is ICANN has this training program that they are going around. In fact, if you're with a ccTLD, they will come to your location and do DNSSEC training for you – for free, actually, if you get a number of people that will go and do that. They'll go and do that. Rick Lamb is somebody who you can talk to about that.

We're also, on the Internet Society side, trying to provide tutorials and documentation work to help ccTLDs with that as well. Some of the vendors are also providing materials as well to help people through that process.

Jacques has some maps he can show you there that will show you some of those things.

KAI HENDRY: Hello there. My name is Kai Hendry. I work for a software company Webconverger. We create an operating system. I'm very interested in integrating a DNS resolver. Your earlier example of the skit with Comcast doing that DNSSEC stuff, that sounded perhaps a better place to do it because you can do it faster and with less complexity on the client side.

I didn't understand, how do I know as a user using an ISP that the ISP's DNS service is DNSSEC enabled?

DAN YORK: Right. How do you know if they're performing the validation? The answer is that you're not going to – what? Yeah, you're not going to know. You're going to go and send . . . You're going to tell them that I want to know what is www.IETF.org and they're going to come back to you and give you an answer that's signed or not. Actually, they're going to give you an answer. You're not going to know whether the validation happened or not, which is why if you do the validation in your app you can know a bit more about that.

KAI HENDRY: I'm assuming if I do it on the client side, obviously more complexity for me.

DAN YORK: Right. Sure.

KAI HENDRY: The thing that I was thinking a lot about is surely this is going to take a longer time to do it. How much longer time is this going to be?

DAN YORK: You mean as far as to do the validation within the application?

KAI HENDRY: Yeah. Are we talking a second or what is it going to be?

DAN YORK: I don't know. As far as the amount of time it would take to do the extra validation inside of an application with a library?

UNKNOWN SPEAKER: There's a few extra round trips that you need to do, but for most of it, it will be in your cache really quickly because [inaudible] pass from the root down via .com and all the TLDs down. You will have cached already. So you only need a little bit of extra records. Some of them you get for free, because when you're asking the query, they come back with the signatures already, so it's not an extra [round trip]. But sometimes you might have to an extra [round trips] to look up the DNS keys and some other records to prove the path all the way to the root.

But in the end, it's very little delay. Of course, for a browser, every millisecond there's a large delay and [you] don't want it, so it all depends on who you're asking.

DAN YORK: Okay. Last question I was – okay. Yes?

UNKNOWN SPEAKER: Hello, everyone. I'm [Awal]. I'm a first-time fellow from Bangladesh. Our organization has a domain [inaudible].bd, but .BD is not yet implemented the DNSSEC. I know that we can have some third parties, so we can get the signatures and everything. We can [inaudible]. But is it secure to do that? .NET, .BD.

.BD is not ready and we want to implement the DNSSEC in our [concessions]. So for [parent], we can choose the [inaudible] secure. Thank you.

DAN YORK: You can still sign your domain underneath .BD even if .BD is not signed. You can still sign your domain. You can still test that. You can do that. The thing is the full validation in the chain of trust can't be done until .BD does that.

There's a great tool called the DNSSEC Analyzer, which is from VeriSign Labs, right? Is that one I'm thinking of? There's two different ones. But they'll show you a nice little chart that will show you with little green dots and little red dots that will show you where the validation breaks down.

So for a while, for instance, one of my own domains, I had signed it on my DNS operator, but the top-level domain wouldn't accept – they were actually signed, but they wouldn't accept DS records from me, which provides the linkage.

So for quite a while, my domain was signed, but people couldn't fully validate it. Then they did accept that I was able to go and do that and it would work on that. Yes, Russ?

RUSS MUNDY: I'd like to add that for those folks that are interested in proceeding with doing DNSSEC, even before their TLD gets signed, I would strongly encourage doing that because one of the things that makes a difference is you're putting a few more steps and processes involved in your overall DNS operation, and it's from just a not disturbing or breaking anything perspective, it's much better to build your system, start doing the DNSSEC signing of your zone prior to anyone depending on it being signed.

That way, if you do have problems with a key rollover or if you do have problems with some version of the name server on one of your authoritative servers not taking DNSSEC, you'll find that out prior to anyone really being dependent on it from a security perspective.

It's a very good idea to start practicing and start running this as early as possible.

DAN YORK: So with that, Julie, do we have any questions in the remote room? No? Okay. I want to say thank you very much for being here, for coming all the way here. We are still around for a few more minutes if you'd like to talk to any of us individually. Please do take a look at these resources that are out there. If you're interested in much more about this, feel free to come back on Wednesday. You can look online and see the agenda to know what we're going to be talking about at different points in time. You're welcome to join us then. Thank you very much.

[END OF TRANSCRIPTION]