SAN JUAN – DNSSEC for Everybody: A Beginner's Guide
Sunday, March 11, 2018 – 17:00 to 18:15 AST
ICANN61 | San Juan, Puerto Rico

UNIDENTIFIED MALE: This is the ICANN 61 DNSSEC for Everybody, A Beginner's Guide meeting on the 11th of March, 2018 from 5:00 to 6:15 in Room 102 ABC.

WES HARDAKER: All right. Welcome, everybody. This is the DNSSEC for Beginners talk, and it's a beginner's guide. At the end of the day, even if you're not super technical, we hope that you'll get some basic concepts of how DNSSEC works to protect the DNS infrastructure. Can everybody in the back hear me? Is the audio okay? Excellent, thank you very much.

We're going to go over DNSSEC and how it works, and we're going to start in the very beginning. And I mean really the dawn of DNSSEC. And it's going to be during the origins of 500 B.C. Is there any way to make those slides bigger? They're pretty small on the back. I'll continue on for now.

I'm going to tell this as a story of Ugwina. Ugwina lives in a cave on the edge of the Grand Canyon. This'll all be relevant in a

minute, I promise. And then I lost my mouse too, when you get done.

UNIDENTIFIED FEMALE: Okay. Try now. That's what I get for helping you out, huh?

WES HARDAKER: There we go. Or at least one of us did that one. Oh, perfect. Thank you very much. I appreciate it. Now hopefully you guys can see that a lot better. The cute pictures of Og. This is Og, and he lives in a cave on the other side of the Grand Canyon.

And it's a really long way down, and Ugwina and Og are good friends, but they don't get to talk that much because it's very hard to get across the Grand Canyon and back.

On one of their rare visits, they note smoke coming from Og's fire, and soon they begin chatting regularly using smoke signals. Until one day, the mischievous caveman Kaminsky – who is responsible for a DNS attack about ten years ago now.

VIKTOR DUKHOVNI: 2008.

WES HARDAKER: 2008. Thank you, Viktor. Caveman Kaminsky moves next door to Og and starts sending smoke signals too. And poor Ugwina on the other side of the Grand Canyon is really confused. She doesn't know which smoke signal to believe.

So Ugwina sets off down the canyon to try and sort out the whole mess, and Ugwina and Og consult the wise village elders. Caveman Diffie – who is named after Diffie Hellman who created one form of cryptography that's used in DNSSEC – thinks that he might have a cunning idea. In a flash, he jumps up and runs into Og's cave, and right at the back he finds a pile of strangely colored sand that has only been found in Og's cave and nobody else's.

With a skip, he rushes out and throws some of the sand on to the fire, and the smoke turns magnificently blue. Now, Ugwina and Og can chat happily again, safe in knowing that nobody can interfere with their conversation. So now they have a magic way of detecting when somebody sends a false signal. This is going to be important later on.

So that was the high-level, cheesy diagram overview of DNSSEC and how it works, and we're going to dive much more deeply now. I'm Wes Hardaker, by the way. I work for the University of Southern California at ISI. And Dan York is the person that normally runs this, but you're stuck with me today because he's

not here this time, unfortunately. And Russ Mundy's going to help me out later.

And most importantly, at the end of the day, you're going to be able to come up and ask questions. So as you're working through the day and you have questions, please write them down and think about them. And we have a whole plethora of DNSSEC experts in the room who can actually help answer questions, anything that you might come up with.

But first, let's talk about high-level concepts of DNS. And most of you probably know how DNS works, at least at a superficial level, but we're going to go over the details in a little bit. The DNS is really structured like a tree. And a resolver knows where the root zone is and how to start the whole process of DNS, and it traverses the DNS tree from the top to the bottom. And each level refers to the resolver at the next level until finally the question has been answered. So the resolver will walk through that tree until it's finally able to answer the question that maybe your web browser started to ask in the first place.

One important thing is that the resolver caches all that information for future use. So if you go back to www.icann.org in a second, the resolver is going to be much faster because it's going to remember that information for a while. So not everything goes up the tree every single time.

But one important thing is when DNS was designed, there was zero security in it. There was no way to tell – it was the equivalent of the smoke signals. You'd get two different answers from two different people, and you'd have to pick one to believe. There was no way to know which one was correct.

And so names are easily spoofed, and once that bad information is in the cache, once somebody poisoned your resolver's cache for www.icann.org, you'd keep going to that same place for a long time until that cache timer expired.

So because the last cheesy cartoons weren't enough, it's now time to do our famous play. How many of you have seen this play before? How many of you keep coming back to watch just the cheesy play and that's the only reason you're in the room? That's what I thought.

So I have some thespian actors here today to help me out. And what they're going to do is we're going to mimic how DNS works. So that tree diagram, we're going to actually show you how that diagram works as we walk through the process. So I have a couple of participants with me.

This is Cathy, she's going to pretend to be the root of the DNS. This is where the top of the DNS starts, the top of that tree. This is the director. Either way, you guys can decide. Can you be right over there? So I can point at people first.

So this is .com. He knows where everything in .com is. This is Warren. And this is Russ, he's going to be Big Bank. I'm going to be a user. And when I need to do some banking, what do I do? I sit down at the computer and I type in, "Hey, I need to go check on my bank balance and see how it's doing today." So I type in www.bigbank.com, and what happens? My browser goes off to the resolver which is located at my ISP. Oh, thank you.

So we even have examples here. That's no the question though. Oh, yes, it is. Okay. Yes, that's the right one. Sorry. We rushed in here at the last second and didn't get quite organized. So I'm going to go to my ISP and I'm going to say, "Hey Mr. ISP, I need to know where www.bigbank.com is so I can go check my bank balance."

UNIDENTIFIED MALE: Hello, Joe. I got your request. You want to go to bigbank.com. But I'm an ISP, I mean a resolver for an ISP and I don't know anything. So I need to figure out where bigbank.com is. The first place I go is to the root. Root, do you know where www.bigbank.com is?

CATHY: I'm afraid not. But I do know where .com is. .com is at 1.1.1.1.

UNIDENTIFIED MALE: Oh, thank you. Fantastic. All right, .com, do you know where www.bigbank.com is?

WARREN KUMARI: Nope, I don't know that. However, I do know that bigbank.com is at 2.2.2.2. You should go and ask him.

UNIDENTIFIED MALE: 2.2.2.2.? Hello bigbank.com. Do you know where www.bigbank.com is?

RUSS MUNDY: Well, as a matter of fact, I do know where www.bigbank.com is. It is at 2.2.2.3.

UNIDENTIFIED MALE: All right. 2.2.2.3. I'll remember that. Thank you. Joe, the address for bigbank.com is 2.2.2.3.

WES HARDAKER: All right. Now I can get to my webpage. And oh, look, I have $1 million in the bank. That is so wonderful. All right, you guys hang out for a minute because we're going to go on a little bit. Oops, where's my clicker? So that's how DNS worked today even

without DNSSEC. It's sort of the equivalent of Ugwina the resolver chatting with Og the server. It's just going on.

The problem is what happens when evil is afoot? I don't know where evil might be. So we're going to do the exact same skit again, but maybe with a problem. One big one and black. So I've decided with my $1 million that I can now afford to buy a house.

So I'm going to go to my computer and transfer the money to my loan agent. I'm going to go to www.bigbank.com, but my browser doesn't know where it is. I'm going to see if my ISP remembers it from yesterday. But it was yesterday, so probably not.

UNIDENTIFIED MALE: What's your name?

WES HARDAKER: I'm Joe.

UNIDENTIFIED MALE: Hi, Joe. Nice to meet you. So you want to go to www.bigbank.com? I don't know where that is. First thing I'll do is go to the root. Hello, root. Do you know where www.bigbank.com is?

CATHY: God, you're a busy man. You know, I really don't. But I can direct you to .com. He's at 1.1.1.1.

UNIDENTIFIED MALE: 1.1.1.1. Thank you. All right. Mr. .com, do you know where www.bigbank.com is?

WARREN KUMARI: Nope. Sorry, I don't. However, I do know that the nameservers for bigbank.com are at 2.2.2.2. You should go and ask them.

UNIDENTIFIED MALE: 2.2.2.2. Cool, I know that place, I think. All right, 2.2.2.2. Hello, Mr. bigbank.com. I want to go to www.bigbank.com. Do you know where that is?

DR. EVIL: Actually, yes, I do. It's at 6.6.6.6.

UNIDENTIFIED MALE: Interesting. Thank you. Hey, perfect. I'll give that to my user. Joe, the address for the bank is 6.6.6.6. Good banking.

WES HARDAKER: Oh, thank you very much. Click, click. Transfer money. All done. All right, so you can see how things go awry, because the reality is that my ISP, the resolver, has no clue whether to believe the signal that came out of the big black cape or not. He doesn't know which one, so he takes the first one he got. And because the evil guy jumped right in front, that was the first answer that the resolver got, and he believed it.

So this is the equivalent to Ugwina the resolver being confused. She doesn't know who the real Og is. And so from a high-level concept of DNS, anywhere in the tree sort of could be poisoned at some point. So we need to make sure that every answer that the resolver gets from the top all the way down is bad. And it ends up looking like there are two bigbanks on the bottom. There's one good one and there's one bad one, and you have no clue which one you get.

So DNSSEC is the solution to all of this, and that has been developed over the last couple of decades. One to get the specifications out, and being slowly deployed. The root's been signed since – you're the expert. It's about six years ago. Seven years ago I think the root's been signed.

UNIDENTIFIED MALE: Root was signed in 2010.

WES HARDAKER: 2010. So DNSSEC uses digital signatures to assume that the information is correct. It's the equivalent of the blue smoke, that nobody else can spoof an answer and confuse you. The key and the signatures are used to verify that the information is stored in DNS as well, so it is stored in the cache as well. It doesn't mean that there are more transactions that have to happen. When your resolver gets that information, it's going to cache the signatures and the real data so that it knows that it'll always be good.

And since DNSSEC is a lookup system, it can simply be looked up like anything else. In other words, the keys are able to be retrieved over DNS as well.

So a resolver knows what the root key is, and this is important because it has to know where to start. It can't memorize the keys of the entire DNS hierarchy, that's way too much information. So it has to have a secure starting point, and it does that with a chain of trust that starts from the root down.

Each level of the key signs the next level underneath until the chain is complete. So you only need the root key, and after that, you can chain down and ensure that the next answer you get is correct and that that next authoritative server will help you chain to the next one. We'll see an example of that in a minute.

That's how the resolver is able to put checkmarks next to the answers so that it knows that the red box is bad because it can verify that that signature was incorrect and you were receiving bad information. And it can keep waiting for the real answer to arrive.

So let's see how this affects with DNSSEC. This is going to be the same skit, the final act with DNSSEC in play. Where'd the question go? Thank you. Wrong question. Do you have stickers?

Okay. All right, so I'm going to do my banking and check on my bank balance to see if my loan went through.

UNIDENTIFIED MALE: [Check the] signatures.

WES HARDAKER: Oh, yes. [inaudible] signatures.

UNIDENTIFIED MALE: All right, so I'm the resolver, and the only thing I know is the key for the root. So I know that the root – I can trust that Cathy is the root, because that's my trust anchored, and my purple sticker is pretty much the same color as her pink sticker.

WARREN KUMARI: So now I should give my signature to Cathy so Cathy can know what my signature looks like. Here you go. You actually have two hands. I was trying to hand you the mic. Anyway, there we go, Cathy. That's what my signature looks like. It's a pretty green color.

CATHY: Yes, I'll go with that, I guess.

WARREN KUMARI: Okay. Now I need to know what bigbank's signature looks like. Hello, bigbank. What does your signature look like?

RUSS MUNDY: Hi, .com. My signature is a bright blue arrow, and I will give that to you so you know what that looks like.

WARREN KUMARI: Excellent.

WES HARDAKER: All right, so now I can go to my bank. Now that it's done all the bootstrapping to get that chain secured between them, my resolver doesn't need to know the entire chain. Note the resolver only got one sticker. That's very important, because there are a

lot of resolvers and a lot of ISPs around the world, and they only need to memorize one piece of information. And everything else works automatically from there.

So I'm going to go to my ISP and say, "Hey, I was trying to go back to my bank today, to www.bigbank.com. Can you help me out again?"

UNIDENTIFIED MALE: Sure. How was your last experience? So I don't know where bigbank.com is. I'm sorry, but I'll go figure it out. So the first place I go is the root. Root, do you know where www.bigbank.com is?

CATHY: You again? Busy man again. I don't know where www.bigbank.com is, but I do know where .com is. It's at 1.1.1.1. But first, I want to tell you that this is what his signature looks like. You better make sure you check that first.

UNIDENTIFIED MALE: Thank you. Let me check that. Let me check mine. Let me check yours. All right, all the colors match. Cool. So 1.1.1.1. Hello, .com. I want to go to www.bigbank.com. Do you know where that is?

WARREN KUMARI: I do not. However, I do know that bigbank.com is at 2.2.2.2. Here's my signature. And bigbank.com's signature is a pretty blue color. It looks like this.

UNIDENTIFIED MALE: So I check 2.2.2.2., I check you, I check this, I check me, check the root, check everything. All good. Perfect, thank you. So I'll go to 2.2.2.2. Hello, bigbank. I want to go to www.bigbank.com Do you know where that is?

DR. EVIL: Actually, yes, I do. It's at 6.6.6.6.

UNIDENTIFIED MALE: Let me check my signature, let me check yours. Hey, come on, that's not the right address. Get. Security by layer. There you go. All right, bigbank, it's me again. I want to go to www.bigbank.com. Do you know where that is?

RUSS MUNDY: I certainly do, and thank you for asking. I have a pretty blue sticker that I'm putting on with this that now matches your star. 2.2.2.3.

UNIDENTIFIED MALE: Oh, perfect. 2.2.2.3. And all the signatures are valid and everything. Perfect, I'll tell Joe User. Hey, Joe, this is the address and I guarantee this is what was in the nameservers.

WES HARDAKER: Last time you didn't do me so well. Are you really sure?

UNIDENTIFIED MALE: Absolutely. Thank you.

WES HARDAKER: All right. Thanks very much. I'm glad you've upgraded your security mechanisms.

All right, can we get our thespians a hand, please? Thank you very much as always, guys. That was brilliant and wonderful.

All right, so let's go on from here. And just like before, now Ugwina is able to check the messages and make sure that the DNS responses are actually correct.

So from here, I'm going to turn it over to my colleague, Russ Mundy, who's going to explain to you why you need DNSSEC, and it's a simple guide to its deployment. So he's going to give you even more details and yet more examples of how these attacks work, and how DNSSEC works to prevent you from being attacked.

RUSS MUNDY: Thank you, Wes. So I'll try to get a spot where I'm not getting blinded, where I can see the slides. Oh, wrong way. There we go. So more of the basic focus of why in the world you need DNSSEC is, who attacks DNS? Nobody attacks DNS because they attack DNS, they attack DNS so they can attack the things that DNS supports. E-mail, chat, banking, you name it. Anything that happens on the Internet. That's why people try to attack DNS, because then they can attack applications.

Today, there are very few applications that are used on the Internet that do not make use of DNS. I'll get these going [inaudible] proper. Here we go. Okay. So what are the results, and how can you do the attacks? Oh, a little – okay, good.

So when you start to log into your e-mail account, you do that from your local machine, but oftentimes when e-mail gets sent off somewhere, sometimes the passwords are in the clear and they can be stolen. That's an easy way to think about it. If you were doing a remote login to a computer across a network, the same sort of thing can happen. There are applications that help with that, prevent the stealing of passwords, but there are still others that allow this to happen.

If you're doing DNS hijacking to wiretap the communications, you can get in the middle and watch everything that goes past.

And again, there are mechanisms that are associated with this that can help prevent this, and if somebody wants more details about how some of these more advanced applications work, I'm going to put a plugin right now for our Wednesday DNSSEC workshop. We have a couple of presentations there that go into great detail about how you can improve your DNS security, your e-mail security and other security today with technology that's out there.

But the thing to remember in terms of simple DNS, it's easily shifted around. We used to do DNS hijacks as part of this, but they actually got a little dull and boring after a while so we don't do them anymore. And one of the things that has happened in the years that we've been doing this presentation is in some places, it's now illegal to do DNS hijacks. So we don't want to have to worry about getting thrown in jail before we leave someplace.

So in terms of tools that are out there, there are multiple tools that have been in the Internet and available for at least ten years that let you just load it up on your computer, sit at your computer and hijack DNS sessions, and then hijack other sessions that the DNS is supporting.

So what is it? You just saw the skit, you've seen the Ugwina. Remember the blue smoke. DNSSEC is like the blue smoke.

When Joe User finally got his answer back from his ISP, as long as it was signed, then that guarantees that in fact, the information that came back had what's called data integrity and source authenticity, communications computerese techy terms. But you know it's right and you know where it came from, that's really what it tells you.

So a little example here with pictures that show Joe User sending out a query. It's dropped down again. My goodness, I'm usually blasting through –

UNIDENTIFIED MALE: [inaudible]

RUSS MUNDY: Okay. So when Joe User sends out his query, it goes to his local recursive resolver, his ISP, and then goes off to fetch the answer. So the question is asked, the answer comes back, and after the answer comes back, then he actually can do what he wanted to do which was talk to his bank. And so this is the pictorial illustration of what's in the skit, and sometimes it's easier to see slides of what occurred afterwards than just to look at our skit again.

So when you go off and do a check like this, sometimes there's an indication that you get back that your connection had passed

the DNSSEC check, and this is an example using a browser that was modified to show you when you've used your DNSSEC checks. And here's a browser that was not modified to use your DNSSEC check.

Now, if you look carefully, what you see is the text on both of them is the same. Why am I not going forward here? Here we go. Dr. Evil is in the middle now. Why am I going back and forth here, Cathy? There, I got it.

Okay, so the question is asked again, it goes out to the recursive server, but oh my goodness, Dr. Evil the hijacker is sitting nearby and said, "Oh, I see a query, and I can give them the answer first." And so that's what he does. But when he does that, instead of the proper website, he's going to Dr. Evil's directed website. There we go. So in the meantime, the other queries are going on and they come back with an answer, but Joe User's browser has already gone to Dr. Evil's website.

Now, with DNSSEC, when Dr. Evil's DNS response comes back, the DNSSEC validation fails, and so instead of going to Dr. Evil's website, he actually goes to the real website. And then he gets the answers back, and okay, here we go. Now, in this case, what we've done is these are some screenshots from a live DNS hijack we did a few years ago.

The browser you see up there is doing DNSSEC. The browser you see on the bottom is not doing DNSSEC, and you'll notice the content on the page is no longer the same. And in fact, what was done in this illustration of how you can use a DNS hijack is we inserted information.

This is the same website with one little difference, that that URL and that DNS name were in there in a way that it was hijacked and the user thinks he's seeing the correct information, and he's not. He's seeing additional information. And this was actually done at the time that Steve Crocker was chair of the ICANN board and everybody knew him then. And most people probably do know him now. And he's saying DNSSEC won't solve world hunger. That was a little bit of humor we wanted to put in.

Now, how many DNS queries are there? I don't know why I keep going, but there we go. That's one page, that's how many queries and responses it took to fill one page. And that's that same page about five years later. You can see they're getting more complex, more DNS queries to fill any given page that shows up on a browser.

So the important thing to remember that what DNSSEC does is it protects the DNS information itself. And without getting into the details of how this is done, what you need to remember is the information in the DNS zone is just as important as any of the

cryptomaterial that shows up in the DNS zone that is being used to accomplish the DNSSEC functions. So the real heart of the question is that the DNSSEC zone data gets validly returned and verified at the user's browser.

Another picture of how the information goes back and forth. This is without DNSSEC. Okay, I thought the next slide was a picture with – okay, a few more words there. So just a few words on doing the implementation.

Back to this picture, if you're sitting here at the Joe User location with a client and you're not running the DNS for yourself, somebody else is running it for you. You need to determine how your DNS service is provided, and as you look at this, some organizations that come to ICANN meetings are very DNS-centric in their business and they have a huge amount of DNS experience and knowledge inside their organizations, and they run it themselves.

Others, not so much. If you are a DNS-knowledgeable, deeply capable organization, you probably ought to be doing your DNSSEC operations yourselves. If your activities are of the nature that DNS is important but you're not DNS experts and you've outsourced it or some other part of the organization is providing it, that's the part of the organization that would be

thinking about, "Gee, should I do DNSSEC there, or should I go to an external provider who does DNSSEC?"

But as you see, the important issue again, protecting your zone data, make sure that that gets validated for users or users can validate it. And however you're doing your DNS operation, just draw the parallel for the DNSSEC operation as it would be an additional capability that you're putting into the DNS, however you're running your own DNS activities today.

So when you put in the DNSSEC information like when we went through the skit, exchanged our keys with each other and then as the recursive server came down, they gathered up all the keys, looked to see that they were right. This very simple illustration up here is where the zone is signed, and down here is where the validation occurs. So that's the additional steps, the additional functionality that you're adding into whatever your current DNS operation is.

So as a rule, however DNS is being operated, if you're a DNS operator and you know how DNS works, you ought to look at doing the DNSSEC signing or validation – or both – in your existing operation.

If you outsourced it and you're not as an organization doing it today, then you need to work with your outsource provider to see that they can provide this to you. And if they aren't able to

provide it to you, I would encourage you to go looking for a DNS provider who can provide you DNSSEC capability.

So that's really our set of presentations, and like Wes said earlier, we've got a group of people in the room that know DNSSEC backwards, forwards and sideways. So we encourage questions, and I'll turn the mic back to Wes.

WES HARDAKER: All right. Thanks, Rus. I think we're done, right? Let me close off with a little bit more information that we haven't created a slide for yet. One thing that you'll note is that DNSSEC will sort of continually have new information around. For some of you who are standing in the back, there are some more chairs here. The table's not reserved for anybody in particular. Please feel free to come up and grab a chair if you want to participate in the question and answer section.

There's current events still going on. Right now, you've probably heard that ICANN is in the middle of a DNSSEC key roll, and so you're probably wondering what that's about. Well, functionally, remember that blue smoke that's in the back of the cave? Every once in a while, it's recommended by cryptographers to change to a different key. And that's really kind of like changing the color of the smoke.

There's a process of doing that to do it in a secure way where most of the world knows about it and so Ugwina realizes you're changing color. And one of the ways that they do that, you can sort of consider it as if you're going to change from bleu to green, for a while you put in both blue and green so that Ugwina can say, "Oh, that's clearly the new color. I should start watching out for that."

That's about the simplest example I can come up with, but that's sort of where we are right now. ICANN through IANA is currently transmitting both blue and green. And so at some point in the future, we will be removing the blue color. And exactly when that's going to happen is up for a little bit of a debate because there have been some technical challenges with actually making sure that everybody gets the new key, because it would be bad for not everybody to get it.

So, does anybody have any questions about anything you've seen today? DNSSEC, deployment, anything that you might have? Okay, please stand up, and then we'll have somebody answer your question. Do we have another mic?

UNIDENTIFIED MALE: Yes.

WES HARDAKER: Andrew, can you – excellent.

LENDON TELESFORD: Hi, good day. I'm Lendon from Grenada, and the question is based on the illustrations presented both in slide and the skit. To be honest, I'm not 100% sure if the question would be framed correctly.

WES HARDAKER: Don't worry about it.

LENDON TELESFORD: First-time Fellow. Based on the illustration, it seems like it's essentially a chain of trust, yes? I didn't particularly see any mechanism – maybe it's hidden – between the client and the resolver. So I'm wondering if there's a way to compromise the resolver. Does the discussion on DNSSEC remain relevant?

WES HARDAKER: Do we have any gold stars? Because that is a brilliant question. And anybody want to take an answer to that one? Viktor.

VIKTOR DUKHOVNI: Sure. Is this on? Do I have to do something? Oh, okay.

So for applications that critically depend on DNSSEC security, you would run the resolver on the same machine that you're running the application, and then there can't be a man in the middle between the user and the resolver. You can forward the queries to ISP resolver, but the validation happens on your own machine. Then you can be sure.

If you're asking the ISP's resolver and there's no resolver that's doing validation in your own machine, then indeed you might be out of luck. But what you do know is that the ISP will not be compromised with stale data that's forged for very long. So if it appears to give you a wrong answer now, if you ask him five minutes later, if it's really him replying, he'll give you the right answer because he was never giving you the wrong answer. It was somebody who's in the middle who happened to give you the wrong answer five minutes ago, but maybe they're not in the middle now, and now you're getting the right answer.

WES HARDAKER: Yes, good. And one more bit of information is that the problem you just described is frequently referred to as the last mile problem, because everything above the resolver can now technically do it, but the client itself often doesn't ask things in a secure way. There are other ways of doing it. One is doing it on the resolver yourself. And thanks, Viktor. Viktor is actually one of

the people who are helping secure mail with DNSSEC and other stuff too, so he's a fantastic expert in the room.

One other note is that the IETF which is actually the body that creates Internet protocols, creates how http works, creates how mail works and that type of stuff is currently in the process of also dealing with DNS privacy and making sure that your requests when you send them to a resolver aren't seen by somebody in the middle. They're defining a secure connection between each client and the resolver.

So in the future, that protection mechanism that's also being used for privacy will likely be able to protect you from other men in the middle as well, so you'll have a secure chain. It'll be a different chain than DNSSEC above it, but it'll still work to effectively protect you. Does that make sense? Okay. Russ?

RUSS MUNDY: One thing that I'd like to add is a lot of people get worried when they see cryptographic mechanisms. "Oh my goodness, this takes a lot of processing and computer power." That's not a problem with DNSSEC. That has been looked at as a factor in the design all along. And I don't have it with me this trip, but I used to carry a cell phone that does DNSSEC on the cell phone itself. So even the small handheld devices can do DNSSEC right on the

device itself. And that's where we eventually hope to get to, is that validation occurs on the end devices, even small ones.

WES HARDAKER: Thank you very much. Next question? Andrew? Oh –

UNIDENTIFIED MALE: [inaudible]

GERARD BEST: Different question though.

UNIDENTIFIED MALE: Go ahead, go for it.

WES HARDAKER: You go ahead and ask your question.

GERARD BEST: Okay. Gerard Best. I'm from Trinidad and Tobago. I'm trying to follow this gold star question here. My question was about open resolvers like 8.8.8.8, and recently heard of the launch of Quad9 – 9.9.9.9 – whether these open resolvers are – just to get some expert perspectives on the usefulness, strength or robustness of those as DNS security mechanisms.

WES HARDAKER: Excellent. Another gold star question. Very good. So the same communication issue getting between you and that resolver still exist, so there's no way to talk securely to Google's server at the moment. I happen to know that there's somebody who can probably answer your question. Right, Warren? Can you comment on – you want to guess where Warren works?

WARREN KUMARI: Hi. Yes, Warren Kumari for Google. So yes, the 8.8.8.8 resolvers are used by millions of users every day, and they do full DNSSEC validation. Actually, the Google public DNS and Comcast were two of the earliest set of large public resolvers which started doing DNSSEC. So yes. There's a bunch of people running them who all take this very seriously, and so I think they're secure. I use it myself, but that's your own decision to make.

WES HARDAKER: Is anybody here from PCH by chance who runs the Quad9? Can you answer whether PCH does DNSSEC?

UNIDENTIFIED MALE: Yes. Quad9 by default does DNSSEC validation, and there's another IP address that we also have that you can use in case

you suspect the DNSSEC validation is failing so that the authoritative servers are – yes, something is going wrong. And it also has DNS over TLS.

WES HARDAKER: Right. Yes, so the nice thing about the Quad9 is that they're also doing DNS over TLS, the same thing that we were talking about earlier that can protect the mechanism, so you know that you're getting the right answer from your resolver as well at the ISP. So excellent question. Any other questions? I think there's one over there.

UNIDENTIFIED MALE: I think you might have been first. Here you go.

RAPHAEL VINCENTE ROSA. My question is that we saw that graph, like how one simple query to cnn.com is resolved, like huge graph. My question is I'm curious about the impact of performance if you have DNSSEC on top of this graph. Or is this already showing the DNSSEC as well?

WES HARDAKER: This was actually created jointly between Russ and I. One of those boxes is my laptop and one is his. And this was actually done quite a while ago. And yes, it actually does show DNSSEC. If

you look really carefully, the orange lines are unsecured. So most of that is actually unsecured, but if you look buried in there, there's a bunch of green. So this was actually fairly early on in DNSSEC deployment, and Russ and I keep talking about doing another round. And we really need to make that happen. So maybe we can work on that later this week, because this graph will look better. There's still a lot of deployment to make happen. With respect to performance, does anyone want to comment? Russ, you mentioned something about that earlier that it shouldn't too much. Have you done any studies with, say, DNSSEC versus what it takes to do a TLS connection over web?

UNIDENTIFIED MALE: So in most cases, the DNS infrastructure data is cached, so by and large, you get slightly larger responses but you don't do more queries. Especially if you're a consumer doing a query through your ISP, you're making the same queries and getting the same answers. It's the ISP who may be doing a few extra queries, but it doesn't affect you very much and they generally have everything you need in the cache because lots of users ask for the same sites.

WES HARDAKER: Does that answer your question, or do you have a – okay. Warren?

WARREN KUMARI: Yes. There's also a follow-on from that where in some small set of cases, DNSSEC actually makes stuff faster, because you get back an answer and some proof which tells you that the answer is correct. In those cases, negative answers – when you make a typo or something – can be resolved instantly, and also some answers which come from sort of wildcards, which is basically a record which says, "Anything maps to this." Those can also come directly from your resolver without having to go to ask all the other [several] authoritative servers. That's a very small speedup though, unless there's an attack in which case it's actually a much larger speedup.

WES HARDAKER: That's a very good point. Thank you, Warren. Okay, any other questions? There's one there in the front, I think.

UNIDENTIFIED MALE: So just in terms of the key exchange that happens between all the different servers, that's happening presumably as a request happens, right? So then the user or the person doing the key exchange has to know where they're talking to. Is that on a delay, or do you cache the keys? How does that work?

WES HARDAKER: That's an excellent question, so I'll go ahead and answer that one. You remember back in the beginning, I did say that keys can be looked up over DNS just like anything else. And so we could walk through ha fairly complex chain of how DNS requests include the ability to transfer the key, and we didn't show that in the skit. The skit would be probably twice as long because you're making more requests. In the process of going to example.com, one of the things that the ISP would do is not just ask for where's www.example.com, but it would ask for what's the key for example.com.

And they would check the signature on that key. So you did see the signature checking happen in the skit. The reality is there are a couple more requests that go in there, and to a large extent they can sometimes be done in parallel too depending on the resolver code. So there is more going on, and there are two to three checks probably as you chain down from each level in the tree. Does that answer your question?

UNIDENTIFIED MALE: Yes. Thank you.

WES HARDAKER: Okay. Any other questions? There's a bunch, actually.

UNIDENTIFIED FEMALE: So it seemed like the security in this whole process was dependent on the little stickers that you handed out at the beginning of the skit, and that was the most confusing part of the skit. How do you organize handing out the stickers?

WES HARDAKER: Excellent, excellent question.

UNIDENTIFIED FEMALE: And also, how often do you have to do that? I think this might have been part of this other person's question.

WES HARDAKER: That's an excellent question. Viktor, I saw your hand go up.

VIKTOR DUKHOVNI: This actually covers something that I wanted to talk about. I was going to ask a question about it at some point myself. One of the things that wasn't mentioned in the skit is that much of the security story really happens not so much between the user and the bank but between the certification authority and the bank, because in the real world – skit aside – your real security depends on the certificate that your bank somehow obtained and whether that certificate is correct or not.

Where the real security happens is when certification authority is giving the bank their certificate. Because how do they know it's really the bank that's getting it? And the answer is they don't. It's all smoke and mirrors. By and large, the certification authorities give their customers the certificates not really knowing who they are. This is called domain validation, and most of you would be shocked to understand how insecure that is.

DNSSEC can help secure that, and increasingly the browser CA forum is adding a little bit of DNSSEC sauce so that domains that are signed with DNSSEC have stronger protection for their certificate issuance. And that then feeds back into protecting the user.

And the reason I'm talking about the CA is that all they do is they say, "That particular party controls a domain." But the people who really know who control the domain is your registrar. You log into an account in your registrar, sign up, buy a domain, it's yours. The stickers that you get are really published by the registrar into the DNS, and you feed the registrar the data over a secure channel. You sign the domain, you tell your registrar your sticker, the registrar pushes it into the registry. There's no extraneous third party. The certification authority doesn't say, "Oh, by the way, bigbank's key is so and so, and this is our wild guess but believe us." Instead, it's the registrar with whom you

have a direct relationship, not some third party kind of pretending to know everything that gets to publish your keys.

Now, you do have to do that periodically as your keys change. If you never change your keys, you never create new stickers. If you roll them periodically, you have to coordinate the key rollovers with your registrar and the IETF is working on that. That's a longwinded answer. There's a lot more to this. I'm not telling you the whole story, but I hope that makes some sense.

WES HARDAKER: So it's sort of the equivalent – as a solid example to follow that up with – he's 100% right – is if you go register a new domain with a registrar – so registrars are things that you actually go register a domain under, GoDaddy being a famous one or DynDNS or a bunch of other ones. When you do that, the ones that support DNSSEC will have a box you can paste your key in in the process of registering, or you can modify it later when you sign your zone. And that's actually what adds that sticker transition. So the owner of a domain would do that during the process of registering their domain. Does that make sense? All right. Russ?

RUSS MUNDY: I wanted to just mention the part of my presentation that said your organization's involvement in the operation of their existing DNS, whether you're under a ccTLD or a gTLD, if you – whoever your organic organization is – are running your DNS, you may want to run all of the DNSSEC mechanisms associated with it. Working through your registrar, getting the key directly into the DNS system.

If you're having your DNS operated for you by, say, your registrar – which is very common – then the registrar themselves can do everything for you. There are recommendations on how frequently the key gets turned over, but this can all be automated, and in most cases, the general view is it's better if it's all automated because there aren't people with fingers on keyboards or forgetting to make a change that's supposed to be put in place. 30 days to six months to a year, depending on what the frequency of change should be.

WES HARDAKER: Many registrars have checkboxes that just say if you're using them for your DNS server, check if you want DNSSEC. And some registrars actually have that checked by default, so you may even have it and you don't know it yet. Let's take another question.

MUJIBULLAH SHAMS: Mujibullah Shams. I'm an ICANN Fellow. DNSSEC signs the queries and responses. I wanted to ask what kind of cryptographic mechanism does the DNSSEC use to sing these queries and responses? And the second question is when a rollover happens, what kind of mechanism is used to share the key between the exchanging parties? Thank you.

WES HARDAKER: Good question. So cryptographically, it's a public-private key mechanism that does the signing, so very similar to how PGP works or how web certificates actually work, and other things that the owner of the key has the private copy that nobody else can have a copy of, and they give out a public key that everybody else can use to validate it.

The length of the key varies, that's up to the zone owner, so it could be anywhere from 512 to 4096 bytes. There is a mechanism for doing elliptic curve if you're familiar with that. So, all those types of things exist. With respect to a rollover, RFC5011 from the IETF defines how to do a rollover mechanism. That's the other magic number series that you'll hear as we're talking about rolling the KSK with the root.

If you own a zone, when you need to change your key, you have to go back to your registrar and update the key. And what happens is that the parent, say, .com will end up publishing

references to both your keys for a while. So you wait a while, and then eventually you switch which key you're actually doing signing. So there's a time period where the rest of the world will notice that you actually have two keys published and either one is essentially valid, and then eventually you go remove the old stuff. So there's a timing mechanism involved with that. Good question though. Warren.

WARREN KUMARI: Yes, and sort of a follow-up from that is that the old key signs the new key currently, and that's what is happening at the moment. It's sort of like perfect forward secrecy type thing, but not really. Wow, I worded that.

WES HARDAKER: That was fine. Okay, next question. We'll do the ones at the table next, okay?

TARAU BAUIA: Tarau Bauia from Kirabati. We are depending on the 8.8.8.8, and now I'm asking if we can skip the ISP and use the Google DNS. Can we have our own recursive or duplicate DNS in our own network? Can that happen?

WES HARDAKER: You can definitely run a DNSSEC validating resolver in your own network if you want. That's a choice of whether you want to use a provider like Google or PCH within the public ones, or Comcast runs an open resolver. Or OpenDNS is another one. I think they do DNSSEC resolution. They do, I'm pretty sure. And OARC has one. So there's a number of open ones that you can use, or you can run your own. Either one of those is fine. And all of the big open ones do run DNSSEC. In fact, they're the ones that do most of the validation. There's a percentage of validated queries that happen, and I don't remember what the percentage is, but most of it is coming from those top four or five resolvers. So let's take a question at the table.

ABDULKARIM OLOYEDE: Hello. My name is Abdulkarim. I want to find out, for example, if you have like – there's this web certificate that you get that for example if you log into like a website and it tells you, "Oh, this website is not secured for some reasons." So what is the relationship between that message you get and DNSSEC?

WES HARDAKER: Excellent question. Does somebody else want to take a turn? All right, so the quick answer is – and it's a good question. They're unrelated for the most part, so when you get the webpage that says you're trying to go to a secure site and we don't trust the

certificate, what will happen is that's actually your web browser checking the website's certificate. In other words, you've already gotten the address, you've already done the DNS, you've gotten the address and then your web browser tried to go to the website and didn't trust the site.

That's actually different. The DNSSEC actually happens before that when you initially need to look up www.bigbank.com and you send it out. And what happens is the ISP – the one example we didn't show in the skit is if the ISP never got a valid answer, he would come back to me, Joe User, and say, "I'm sorry, I found nothing." So he wouldn't accept any of the answers. And then you would get "site not found" or "host not found," which you've probably seen on other sites. If you type in random gobbledygook, you don't end up at a good webpage, you come up with a "host not found." And so you get a different error.

There are some more complicated examples of what happens – your web browser actually understands DNSSEC and all that kind of stuff.

ABDULKARIM OLOYEDE: So the certification is like you have two sets of certifications? One is –

WES HARDAKER: Yes.

ABDULKARIM OLOYEDE: Okay.

WES HARDAKER: Yes, they're very different paths for how you do certification. It's best if you have both.

ABDULKARIM OLOYEDE: Okay.

WES HARDAKER: Viktor?

VIKTOR DUKHOVNI: Where they overlap – much too loud – is that your certification authority that issued that certificate in many cases used insecure means to verify that the site is who they say they are. They might have used unsigned DNS and somebody appearing to modify a record, a challenge it's called that the CA says, "Okay, prove that you control this site. Publish some data in the DNS. If we see it there, we know it's you." But it could be a man in the middle. So if your traffic is rerouted through some country in a BGP hijack attack, all of a sudden it goes through an

unexpected path, that attacker might be then obtaining certificates for various sites that the traffic would not normally go through them for. So really, the overlap is that the CA validation of domain owners is very insecure. It's not very often attacked. That's the only reason we're safe, is that by and large, these attacks are infrequent. But they're certainly possible.

WES HARDAKER: And a fully deployed DNSSEC world would solve that problem.

VIKTOR DUKHOVNI: Right, provided the CAs then also use DNSSEC to validate.

WES HARDAKER: When I say fully deployed, that includes validation. That's a good point. All right, there was another question at the table. Somebody else had one.

LAUREN BURKHART: Yes. My name is Lauren Burkhart from the United States. I have a very basic question and absolutely zero technical knowledge before yesterday. You mentioned basically you can essentially sort of check the box with your registrar, whoever might run DNSSEC for you. How prevalent is this, and why wouldn't you

check the box? Or when you referred to this being fully deployed, where are we in that?

WES HARDAKER: That's an excellent question, and actually, one of the best people that monitors the levels of stuff is sitting right there. He runs a number of things. There's actually another – I'll give one reference before I turn it over to Viktor who monitors one half of things. There's a program called SECSpider, and you can go search for it, "SECSpider." That's actually a database of trying to find out all the domains that have been signed. And you'll find that the percentage is not as high as you want. And if you look at .com, it's like .5% or something.

UNIDENTIFIED MALE: [inaudible]

WES HARDAKER: Yes, if you go to the DNSSEC workshop, they often list those [inaudible].

UNIDENTIFIED MALE: [inaudible]

WES HARDAKER: Yes. But .5% of .com is actually pretty big in numbers, it's over 500,000 sites. But it's not everything, and there are technical reasons why some major companies haven't deployed it yet because of the tricks they're playing with the DNS to do load balancing and stuff. Viktor.

VIKTOR DUKHOVNI: So I have statistics. I do daily scans. I'm currently tracking 5.2 million DNSSEC-signed domains. Most of them are in northern Europe, so especially Holland, then Sweden, Germany, Norway, those kind of places. A bunch in the U.S.

WES HARDAKER: Czech Republic.

VIKTOR DUKHOVNI: Czech Republic. The rest of the world, not so much. The distribution of DNSSEC is very uneven. So .com has two thirds of a percent, 813,000 domains are signed. Sweden, more than half of the domains are signed. Holland, more than half of the domains signed. .bank, every single domain is signed, but there's a tiny number of domains and most of them are just parked. The banks register their trademark and then never use them. The same with .insurance. There are 192 .insurance

domains, every one of them is signed, I don't know that any of them are in use.

Overall, the number is between two thirds of a percent and 1% at the moment. But in some places, the use is very high. In the Netherlands, DNSSEC is cheaper. The registrar will give you a discount for DNSSEC-signed domain. Everybody gets a DNSSEC-signed domain. The same in Sweden.

WES HARDAKER: Yes, excellent points. As far as the TLDs go, like 86% of the TLDs are signed, but a lot of those – to get one of the new gTLDs, you had to support DNSSEC. Over 50% of the countries I think country codes are signed. .com, .net, .org, .mil, .edu, .biz, .info, those are all signed. So there's a large number of top-level stuff that's signed. We're trying to push down at this point and get more per company. Please follow on.

LAUREN BURKHART: Yes, is there a simple to explain reason or set of reasons why you wouldn't want to sign? Or is it just a matter of time and work that needs doing?

WES HARDAKER: Russ?

RUSS MUNDY: It's both. Okay, as Wes mentioned earlier, there are some organizations that have taken advantages of various things that you can do with DNS to help them from their business perspective, and the things that they're doing won't work if they actually follow the DNS rules. And when you start to apply DNSSEC to them, it identifies that something's broken.

So there are instances of that nature. There's also – I think one of the biggest problems is one of education, which is one of the reasons why we do these sessions. We want to be able to familiarize people with the basics of what's involved in DNS security so that we have more people in the world that have a general understanding and who go back to their organizations and their jobs and start asking questions. "Why aren't we doing DNS security? What can I do to help this move forward?" That's a very important function of this particular session, and thank you for asking.

WES HARDAKER: Jacques?

JACQUES LATOUR: Jacques Latour. I'm with .ca and overall, we have 2.7 million domains in Canada and we have 1000 signed with DNSSEC.

That's a very low percentage, it's pretty close to zero. That's zero dot something. And the challenge is today – this is what we're dealing with. The registrars that we have, some of them support DNSSEC, and the registrants need to pay more money to enable that. So that's a challenge. You have to pay more money to get security, and often it's a game for the lowest domain, so people just don't buy DNSSEC for .ca. So money is an issue for .ca for implementation.

WES HARDAKER: Okay. All right. Andrew.

ABDALMONEM GALILA: Abdalmonem from Egypt. I don't trust anyone, by the way, so I don't trust my [ISP] So I will make my local DNSSEC validator at my machine. So I think it is better to be away from man in the middle attack between me and my ISP. So, why not ICANN or you advise everyone to have local validator at their machine?

WES HARDAKER: That's an excellent question. Viktor?

VIKTOR DUKHOVNI: Sure. If your machine is at all mobile, if you carry it into airports and hotels and so on, if we're talking laptops, unfortunately,

many of these environments are very hostile to DNSSEC because the infrastructure they have, the captive portals that get you to log in, accept their terms and so on have DNS lies in order – no matter what domain you type initially, say, google.com and instead you end up at the hotel's portal. Or they just have technology for one reason or another to enforce their policies that modifies DNS answers.

So laptops are mostly today unable to directly use DNSSEC. It's just cut off too often. If you have a server in a datacenter, absolutely deploy DNSSEC locally on your machine. I'm surprised operating systems don't yet do that by default. They increasingly will.

WES HARDAKER: Okay. Russ?

RUSS MUNDY: Yes, that's a real good point you bring up, Viktor. There are various ways that one can take care of problems of that nature, and there is a software program called DNSSEC-Trigger from NLnetLabs that was designed specifically for these types of situations, and it has a set of mechanisms built into it that works very hard to use various connectivity paths to actually do DNSSEC.

And if I remember correctly, for those kinds of environments that Viktor was just describing, the first query or the first connection has to get set up to the hotel because they are blocking your path if you don't first talk to them. But after that, then it starts doing DNSSEC because you can run it after that. Personally, I do it a little differently. Once I get the okay with the hotel blocking server, I just turn on my VPN and go to DNS servers that I know and do DNSSEC that way. So there are various mechanisms to get around it, but it's not always all that easy to do.

WES HARDAKER: It's getting better. Tools like DNSSEC-Trigger and the fact that DHCP is often handing out the site that you need to go to log into so if you're on a modern operating system, sometimes you get a little popup that says you need to log into this network. Those types of things are actually helping get away from the captive portal issue so that you can do the network setup and actually have DNSSEC set up locally.

Right now, that's sort of the expert-level thing, and I encourage you to do it if you're an expert and you can pull it off. I do, but it's hard. We're working toward that. Some Linux software vendors actually do turn it on by default in their software config, but there are only a couple of those. And it's creeping out slowly,

so I think it'll happen. Do you have another question, Andrew? Back over there?

UNIDENTIFIED FEMALE: Can I Spanish?

UNIDENTIFIED MALE: I don't know. I don't think so, because he doesn't have a translator. Unless you're someone – you can ask right now if you can ask questions in Spanish.

FRANSLEIDY DE JESÚS DÍAZ: My name is Fransleidy de Jesús. I'm from NextGen. I can do a question in Spanish?

WES HARDAKER: Does somebody have translation, or does somebody know Spanish really well? My Spanish is awful. Do you have it? Yes, okay, go ahead.

FRANSLEIDY DE JESÚS DÍAZ: One of the problems is that when the information from the DNS arrives, we do not know if the information has been forged. That's why the root zone has been changed to KSK. That has been one of the big changes in February. So when you actually

do the rollover, what are you doing for that problem to not happen again?

WES HARDAKER: We'll wait one second for the translation to catch up, and then Jacques will answer your question.

JACQUES LATOUR: So I think that the question is we're rolling the key because the one that we have today could have been compromised. That's not what happened. Is that your question?

FRANSLEIDY DE JESÚS DÍAZ: Because in February when you doing the Los Angeles for change the key for DNS, the [inaudible] the KSK, one of the problems in this presentation is the information, you [don't know], see if the information is [wrong] or they have more information because you changed to the [inaudible]. I don't know [if you can explain].

JACQUES LATOUR: So in February, we changed –

WES HARDAKER: She's talking about the generating of the new key last [inaudible].

UNIDENTIFIED MALE: Just a quick – I'm happy to translate.

WES HARDAKER: Yes, go for it.

UNIDENTIFIED MALE: Two things. So there was an event in February, so maybe you could talk about that, how it relates to the KSK. And then her second question was, what are you doing in regards to the KSK to make sure it's successful? So what was the event in February, what was the thing that got pushed to later this year, and what are you doing? So just kind of three different things. Thanks.

WES HARDAKER: Okay. Russ?

RUSS MUNDY: So there is a well-documented, extensive process that describes how the KSKs are generated, the first ones, and how the new one was generated. Part of that process is there's independent people that had been previously identified as community support people that come in and actually witness to see that all of these steps are probably done. It's all done in a highly secured area.

And the key storage, once it's generated – well, it's generated in a hardware device, it's kept in a hardware device, and that hardware device is literally locked up when it's not being used. So that's how the current KSK was generated and how the new KSK was generated. So the public part of the KSK is out and visible and available for anyone, and hopefully everyone to see and make use of.

The private part is extremely tightly guarded and secured. And when it's changed, all of the key change ceremonies are taped and are available. Anyone at any time can go back and look at those. They're all available through the ICANN website. And so there's an ISO 9000 base standard that was used to create the whole facility and it has been audited I think twice or three times by independent auditors. There's a lot of effort that's gone into making sure that the private key is in fact very carefully protected and safeguarded.

And the reason for the change has nothing to do with any kind of compromise. It's a change because the procedures associated with the root has said we will change this every certain amount of time. And that's the reason for the change, not because there's any kind of compromise. Does that help with what your question was?

WES HARDAKER: Let me add one more point, which is that operationally, it's a good idea to go through that process and change keys even if you don't have a problem. It has to make sure that everybody can pick up the new key, and it's better to do that in a time where there is no problem. Because if you have to rush through that process and you've never done it before, your chances of getting it wrong increase.

And so your other question on timing, so the generation of the new key is what happened in February last year. The changeover to the new key was supposed to happen in October. Due to the fact that we were measuring whether people were using the new key or the old key, we found that there were still a lot of people that haven't started switching to validating with the new key yet, so ICANN delayed that and there's currently a call for community input on the timing proposal to now delay it until next October.

So there are opportunities for anybody who thinks they understand this well and want to have an opinion about when we're going to change to the new key. Now is the time to go read that document and actually provide public comment, because that's open right now until April 4th or 5th, I think. 3rd? 3rd. Okay. Is there another question? We have probably time for I think one more. Yes.

ABDALMONEM GALILA: How long time frequently the root key root KSK rollover will take?

WES HARDAKER: How long will it take?

ABDALMONEM GALILA: Yes.

WES HARDAKER: Good question. It was originally designed to be done – until we delayed it – to be done within five months.

UNIDENTIFIED MALE: [inaudible]

WES HARDAKER: If you know the timing, speak.

UNIDENTIFIED MALE: [inaudible]

WES HARDAKER: Yes. The reality is that there's no –

UNIDENTIFIED MALE: [inaudible] like how many years?

ABDALMONEM GALILA: Yes, how many years.

UNIDENTIFIED MALE: The question is about frequency, how frequently is the key rolled?

WES HARDAKER: Okay. Right, so how frequently. That's actually subject to debate. I think that the current ICANN-published policy says every five years right now. Whether it takes another five years or whether they do it more frequently, my guess is that after we finish rolling this one, that policy will be reexamined and updated as to whether they want to stick with that or change it. So I think right now it's every five years. The reality is I expect that policy to be reviewed again after this goes through.

UNIDENTIFIED MALE: Another one.

WES HARDAKER: Follow on real quick.

ABDALMONEM GALILA: Last question. Does ICANN have an authority to restrict ISPs to do DNSSEC validation and to update their DNS software to accommodate or roll the key, or to get the KSK from the root zone automatically to complete the task as trust anchor?

WES HARDAKER: There's nobody who has authority over all the ISPs in the world, so no. That's not possible. In theory, governments could do that, but there's no government that has tried to mandate ISPs do that. Good question though, I've never had that one asked before.

All right, thank you all for coming. This is the end of the timeslot so I have to let you go, but wonderful set of questions, and hopefully you learned something today. So thank you.

[END OF TRANSCRIPTION]