BUENOS AIRES – DNSSEC for Everybody
Monday, November 18, 2013 – 17:00 to 18:30
ICANN – Buenos Aires, Argentina

JULIE HEDLUND: Welcome everyone to DNSSEC for Everybody – A Beginner's Guide. Please, as you're coming in, take a seat at the big rectangular table. We want you all up front so we can involve you in our wonderful activities today. Trust me, it will be lots and lots of fun. Please come on up. We'll begin momentarily. We're trying to fix something on the camera. What we'll probably do is get going on the first part shortly. I'll get the slides showing and then we'll get going.

PRESENTER: Hi everyone. We're going to get started. We were originally going to be filming this for the camera, but we seem to be having technical issues so we'll skip that for now. The camera issues we're fairly sure are not DNSSEC related, but some of this stuff is complex, so hey, it might be. This is the DNSSEC for Everybody Session, otherwise known as DDNSSEC for Beginners. This DNSSEC stuff can be kind of intimidating. It's fairly complex and technical. It has lots of buzzwords in it and stuff like that.

We're not going to be able to make you DNSSEC experts, but they should at least provide a general introduction to DNSSEC. At the end of this you can at least use some of the buzzwords and people won't laugh too much. Before we get started, let me introduce some of the people that are going to be participating in the skit that we have.

We have Russ Mundy, who is the principle network scientist for Parsons. His group also does the DNSSEC tools and he's been involved in this for a long time. He's ben organizing many of the workshops and things like that. Sitting next to him is Roy Arens. Roy is research fellow and Nominet, who is the .uk registry. Roy was also instrumental in creating a bunch of the DNSSEC specs, he wrote some of the RFCs, etc.

Next we have Julie. Julie helps organize this and keep this somewhat truthful. Further down is Norm. Norm is the director of DNS intelligence for Crowdstrike. They're a cyber-security company. Who am I missing? Jaques, who works for SIRA, who's the .ca registry, and I think that's everybody. Somewhere we also have Dr. Evil, but nobody knows where he is.

DR. EVIL: [Evil laugh]

PRESENTER: Oh dear. How many people here know anything about DNS? If you could just stick your hands up? Okay. How many know anything about DNSSEC? Okay. Well, I'm sad to tell you that whatever you know about the history is probably wrong. Yes. You probably think it was invented 10-15 years ago. Actually, it was originally invented around 7,000 years ago and it was invented by a bunch of cavemen.

This is one of them, a cavewoman. Her name is Ogwina. She lives on the edge of the Grand Canyon. Here's another one of the inventors. His name is Og. He lives on the other side of the Grand Canyon. Ogwina

and Og have a bit of a thing going. Let's call it that. Unfortunately, it's a long way down the Grand Canyon and a long way around, so Ogwina and Og don't get to see each other nearly as much as they'd like to.

On one of their rare visits, as they're looking lovingly into each other's eyes, they notice the smoke coming up from Og's fire. Soon they figure out that they can use smoke signals to communicate with each other, and that way they can send love letters back and forth and arrange trusts and lunches and things like that.

Then, one day this mischievous caveman called Kaminski moves in next door to Og. Kaminski thinks it's kind of fun to insert random messages in-between the conversations of Ogwina and Og. Og will be telling Ogwina how much he loves her, and Kaminski will start talking about bananas, or something completely unrelated. Poor Ogwina, she gets really confused. She has no way of knowing which are the correct messages and which are made up ones being inserted by Kaminski.

She gets annoyed with Og one day. She goes all the way down the Grand Canyon, takes the three-day hike across, goes all the way up the other side, and Ogwina and Og try and figure out what they can do. They go along to the village elders and they ask them for advice. They want some way they can chat without k1 getting in the middle. One of them is a caveman called Diffy. He sits there, he thinks and he ponders the problem.

Suddenly he has a brilliant idea. He gets up and he runs into Og's cave. In the back of his cave there's this really special sand. One of the things that makes the sand special is the fact that it's only found in Og's cave.

Diffy picks up a big handful of it and he runs out and throw is into Og's fire. His fire goes this brilliant blue fire. Now Ogwina and Og can carry on their conversations quite happily. Ogwina knows that all she needs to do is only listen to the signals that are blue, because only Og could have possibly colored them blue. K1 can sit there and try and insert messages, and Ogwina knows she can just ignore them.

If you take one thing away from this little session, it's that what DNSSEC does is provide blue smoke for DNS. It provides a way for the recipient to know that the person who sent the message is the one who looks like they sent it. Only one person can insert the blue smoke, and so the recipient can determine which are the correct messages and which ones aren't. With that I'll hand this over to Roy who will do a little bit of a presentation with the DNSSEC part.

ROY ARENS: Hello everyone. My name is Roy Arens. I'm going to talk to you a little bit about DNS, a little bit about DNSSEC. First I'd like to have a show of hands. I know we did this before but I'd like to phrase the question slightly differently. Who understands that we have root servers in the world? Okay, that's good. Who understands on a basic level how DNS works? That stuff is delegated from the root down, etc? Okay.

This is your standard DNS tree. We have the root on top. The root delegates stuff to .uk, .com, .ar, and under that you have second-level domains – nic.ar, etc. An ISPs resolver or your company's resolver basically knows where the root servers are. Joe User, and we'll introduce him later, has no idea what the DNS looks like. Joe can only

talk to the ISP. The ISP will then go from the root, all the way down to the level you want to be at – for instance bigbank.com.

The resolver gets the address and that information is cached for future use. At this moment I would like to show you how we really do DNS. This is a small skit, a small play. We have a few players here. The slide you saw before had the root servers, it had .com servers, it had bigbank.com, etc., etc. While the team is sorting out their props I'll introduce you to Joe User. Norm is going to be Joe User. He's your typical user. He's sitting on his laptop on his system at home trying to browse somewhere.

He types in something in the URL bar, in the address bar of your browser, and everything that we'll show you here, basically happens after you type in "enter" and before you see the page. It happens in milliseconds here, but we'll show you what happens under the water. To be fair, that happens with computers and systems, not with real people, because then we'd be much faster.

We have Joe User over there. We have our ISP over here. We have the root server over here. I am .com and this is bigbank.com. What you'll see now is how typically a resolver acts. I'm going to hand the mic over to Jaques Latour.

JOE USER: I'm doing some Internet banking. I have to pay my bills. I don't know anything about DNS and I don't care. The only thing I know is every month I have to pay Mr. ISP for my access, and he handles everything. I

have to pay the bill. I sit down at my computer and I type in www.bigbank.com.

ISP: Thank you Joe User. I'm an ISP. I've just been turned on so I don't basically know anything. The only thing I know is where the root is. That's the only thing I'm programed to start with. I don't know where the bank is, so I'm going to go and ask the root. Do you know where bigbank.com is?

ROOT: No, sorry, I don't know where bigbank.com is. I do however know where .com is. .com lives at 1.1.1.1. You should go and ask him.

ISP: Thank you very much. Hey, .com! I'm looking for bigbank.com. Do you know where I can find that?

.COM: No, I don't know where www.bigbank.com is, but I do know where bigbank.com is, and it loves at address 2.2.2.2.

ISP: 2.2.2.2. Hello, bigbank, I'm looking for www.bigbank.com. Do you know where that is?

BIGBANK: Well, as a matter of fact, yes, I do know where www.bigbank.com is. It's at 2.2.2.3.

ISP: Thank you very much. Hey Joe, the address is 2.2.2.3 for bigbank.com. Good luck.

JOE USER: Awesome. So now I've typed in the web address for my banking site. ISP's given the number back and away my computer goes so I can do my banking. That's basically how a DNS transaction works. It does it a bit faster in real life, but that's the basis of a transaction. Yes, now he's cached the answer, as you can see in his belly.

Now what we're going to do is show you what a man in the middle attack looks like. This was actually the motivation behind DNSSEC. We'll show you what that attack looks like, using these wonderful actors again. We'll do the same scenario over again this time, with a man in the middle attack. Here we go. I'm going to sit down and do some more banking. Bills, bills, bills. I type it in: www.bigbank.com.

ISP: Thank you Joe. You want to go to Bigbank? I need to figure out where that is. I've got to go and ask the root first. Do you know where www.bigbank.com is?

ROOT: I'm sorry Mr. ISP. I don't know where www.bigbank.com is. All I know about is .com. They're at 1.1.1.1.

ISP: Thank you. Hello Mr. .com. I'm looking for www.bigbank.com. Do you know where that is?

.COM: I don't know where www.bigbank.com is, but I do know where bigbank.com is. Bigbank.com lives at 2.2.2.2.

ISP: Thank you very much. Hello bigbank. I'm looking for www.bigbank.com. Do you have the IP address for that?

DR. EVIL: Oh yes, I do. The address is 6.6.6.6.

ISP: Beautiful, thank you very much. Joe, here's the address. 6.6.6.6. That's where you want to connect and do all your online banking.

JOE USER: The idiot. Thank you Mr. ISP, that's great of you. Again, I think I'm going to bigbank.com. Dr. Evil here has injected his own address and now he owns me. My banking is now going to Dr. Evil. He's got all my $3.55c. What I was talking about earlier is this is how DNS works without

DNSSEC. There is such a thing as a man in the middle attack, and it has happened to very large banks.

But part of the problem here is there's no element of knowledge between the different servers here in the tree; in the hierarchy. They don't share information, they're not talking to each other, they don't even know each other.

If you remember back to the comic strips of the blue smoke, there's a concept called the chain of trust. This is where the authoritative servers share information and they share keys, but that's like the blue smoke that they're going to be sharing. If we go through the chain of trust…

ROY ARENS: Before we go to the next skit there's some stuff I need to tell you. Remember the slide where we had Ogwina on the left hand side, chatting with Og? In DNS terms, on the left hand side, the beautiful lady over there that's Ogwina, she's the resolver. Ogwina is a modern woman. She's not just chatting with Og, the server, but with many different Ogs – many different servers.

As you know, Ogwina doesn't really know who the real Og is. The resolver, in this case, doesn't really know what the real servers is, because all this information that goes back and forth might have been spoofed. With DNSSEC – we'll show you that in a second – Ogwina is now able to distinguish between a false Og, or a false server, false information, and the real Og.

She does this with blue smoke. In the DNS world we do that with DNS keys and signatures, etc. In DNS there's really no security. This protocol was invented around about 1982, 1983. This was way before the web even existed. At that time, a whole lot of researchers, a whole lot of networks were coupled together, but really no one thought about this being abuse. In this case, there's no security, these names are easily spoofed and since DNS has this concept of caching this spoofed information will not be cached. It's called cached portioning.

We'll do the skit again in a minute. In this case we had root and .com, the original bigbank.com on the left and the false bigbank.com on the right. That's why it's in red – because red is bad. How do we do this with DNSSEC? DNSSEC uses the concept of digital signatures. Basically, you create a key pair where there's a public key and a private key. The public key is something you can give to anyone you like. The private key is something that you store somewhere very secret and safe.

The public key is basically nothing else than a bunch of bits, and since you can store anything in the DNS, like an address – if you want to look up an address and ask for type a – you can also store keys in the DNS and that's what's we call a DNS key. The public key is something we can store in the DNS. When you sign DNSSEC you create something called signatures. You do that with a private key, so the world who has public keys can validate or verify that.

These signatures are nothing else but a bunch of bits that can be stored in the DNS as well. Now we have DNS keys in the DNS and now we have signatures in the DNS. There needs to be a link – and I'll show you that

in a second – between all these entities; the root, .com, bigbank.com. Basically the glue between the root and .com, and the glue between .com and bigbank.com is something called the DS record. That's nothing else than a simplified version of the DNS key.

The root's DNS key, the one that everyone has, needs to be in a resolver in order for the resolver to trust information coming from the root. Since the resolver now trusts the root, the root has signed information about .com so the resolver can now trust .com as well. .com has signed information about bigbank.com, so the resolver now can implicitly trust bigbank.com as well, and so forth.

In that same high-level concept, since we've signed all this information and we have this chain of trust from the root to .com, to bigbank.com, we can now properly distinguish between false information, which cannot be properly signed, because the attacker does not have the private key that bigbank.com has, the resolver can now properly distinguish between properly signed information and unsigned or falsely signed information.

Now we do the skit with DNSSEC in place. Everything I just told you, we're going to apply that to the same skit that we did before. Can I invite my friends over again?

ROOT: Hello everybody. First off the root needs to be signed. I will sign myself. Hey everybody, this is my key. Now I need to exchange keys with .com. Hello, are you really .com?

.COM: Yes, I'm really .com.

ROOT: In that case we can exchange keys.

.COM: Thank you.

ROY ARENS: This is a key exchange, basically. What we just did we're going to do again here.

.COM: Hi, I'm .com. Can I trust you to be bigbank.com.

BIGBANK.COM: I am bigbank.com.

.COM: Perfect. You deserve a star. Well done Russ.

ISP: Now I have all of my zone signed and so the address for www.bigbank.com is signed.

JOE USER: Now we're going to do a DNS transaction again, same thing, again with the man in the middle attack, but this time we have the DNSSEC to back us up. Same transaction, more bills. I don't actually have any money left because I got ripped off last time by Dr. Evil. Banking time. www.bigbank.com. Mr. ISP?

ISP: Thank you. You want to go to www.bigbank.com? I don't know where that is so I'm going go ask around. Now I need to talk to the root. IANA published the public key for the root and they validate that, and I know he's the root. Do you know where bigbank.com is?

ROOT: Yes, I do. Bigbank.com is at 1.1.1.1. Give me a second, let me sign this for you.

ISP: All right. I checked the signature you gave me. The signature matches your key. We're good. Thank you very much. I'll go talk to 1.1.1.1.

ROOT: When you talk to him his key is… Here's a really long number.

ISP: Thank you. 1.1.1.1, hello .com, I want to go to www.bigbank.com, but I don't know where it is?

.COM: I don't know where www.bigbank.com is, I do know where bigbank.com is. Bigbank lives at address 2.2.2.2. Here's a signature over that information, and here is also the key for bigbank.com.

ISP: Keys verified, signature, check… It's all good. All right, I'm good. I'm going to ask bigbank.com for www.bigbank.com.

DR. EVIL: I know the answer. The answer is 6.6.6.6.

ISP: Thank you very much. Let me check the key. The key doesn't work… Hey! Get the hell out of here! Thank you very much. The address is…

BIGBANK: The address is 2.2.2.3 and it's signed.

ISP: I need to verify that… Perfect, thanks very much. Hey Joe, I've got the address. It's 2.2.2 and the signatures are valid.

JOE USER: Great. Thank you very much ISP. Now I have a validated resolve to this number and that's how the transaction works. As an important point

here, you'll notice that me, as Joe User, didn't have to do anything. I just asked the same request and it all happened up with the servers, the root and the ISP, the registries and registrars and the domain owners. But it doesn't put any onus on the users to do anything.

That's it. Thank you. Dr. Evil, you have to really like that.

PRESENTER: Well, thanks everybody. We really do try to make this the most fun session of the week. I don't know if we succeed or not but we sure try. We do indeed want to keep it informal, so there hasn't been any questions raised yet, but this is a very informal session so if anything comes to mind that you want answered please holler, raise your hand, whatever, and we'll try to stop and take care of it then or tell you exactly where it's coming from.

My part here in the presentation of this session is to do two things – to give a more specific example of hijack and what can be the actual result from an actual one.

JAQUES LATOUR: I already have a question based on the little play you just had. I saw a person that had a strong French accent doing all this in English. Is it the case that you really need to use English language for DNSSEC or does it also apply to internalized strings?

ROY ARENS: Thank you. It will function, and it was designed to work with any data that's contained in the DNS. If it's IDN data, any other character set, from a DNS perspective it's just bits. It will work, if you have other languages that you're using in your normal processing. Great, thank you. Let's go to the first slide because we had our skit up here about DNSSEC and you saw why you might want to worry about it.

The bottom line of all of this is that the DNS content itself needs to get correctly to the applications that are going to use it, or bad and unexpected things are going to happen. It really is the end-user application that you're the most worried about. End users really think at most about the DNS, it's just the names.

But the network infrastructure underneath, and the mechanics of it, all use IP addresses of one sort or another, so you've got to get this translation correctly done. That's where the hijacking takes place, in terms of changing the actual IP addresses you're going to use for the mechanisms that are going to move the bits around. That's really another way of describing what a DNS hijack is.

When you do a DNS hijack it's not really because you want to hijack the DNS for the DNS's sake. Nobody really cares. The DNS is important to DNS geeks, but for people doing real work on the Internet they care about their applications, and that's what the people are after when some type of hijack is done. So one of the things that I found surprising – about five years ago when I stumbled across this – there is openly available software that will let you do DNS hijacking available on the Internet.

I found a couple of universities that actually required, as part of their computer science course, for the students to write software that did DNS hijacking. Talking about it and… I don't know how much ethical usage things they'd associated with it, but I didn't see much, but I thought it was unfortunate that they were really pushing to write code that actually did this. There's a lot of code in existence out there and it's something that college students can obviously do, because they've been asked to.

When you're actually going to make use of DNS and DNSSEC on a website, you can use the information that's part of the protocol to do things that will help people see whether or not they're making use of DNSSEC when they get to those websites. It's not done very frequently but there are a few sites that do it, and of course some of the sights associated with… Are projects that do have this capability.

What we did is we modified… Part of our website is so that it can actually show a real, live demonstration of a hijack of part of a webpage. This slide shows how the actual packets are flying around, which is a different picture, but more or less the same thing we just showed you with the skit. What happens, Joe User sends his query to his local ISP, local resolver, and it starts and does its back and forth across the network, and eventually gets the answer back.

After the answer returns, that's when Joe User's machine is actually able to talk to the bank. The application can't really do what the user is wanting to do until it gets its complete DNS answer returned. When you

have a DNSSEC-aware browser, the particular site I was describing earlier is DNSSEC-deployment.com.

You can see in the top browser where the DNSSEC check mark is on, and we're able to detect that by watching the DNS queries come into that particular website, and if it has the DNSSEC capabilities and it's asking the right questions and properly processing the answers. We'll give him a DNS checkmark. If your browser does not have that, you'll get a "DNSSEC is off". Just a little indicator to tell the user that's looking at the page whether it's on or whether it's off.

The equivalent of Dr. Evil is our Dr. Evil hijacker here. As the query is sent by Joe User, Dr. Evil observes through some manner that he's able to do, and if you're in the right place it's pretty easy to do, what the request is. So what Dr. Evil does, number two, he gets the answer back to Joe User prior to the real answer returning. You can see in this illustration it goes a lot further. The actual answer takes longer than the fake answer that gets back.

Now, if you're using DNSSEC, just as you saw in the skit, what happens is Joe User is able to detect that the answer he gets from Dr. Evil hijacker is invalid, and so he ignores that answer and then he'll get the proper answer from the actual location that's giving him a DNSSEC answer, and then he can get to the web server that he really wants to get to.

It's another way of illustrating what you just saw in the skit. I didn't have all of root and all of .com, because if you put all of those machines in it gets might busy on this slide. Are there any questions that anybody has? Yes?

HOSAM HASSAN: I'm from Egypt. Actually, I have a question and I have written to [inaudible 00:39:30] cannot get an answer for my question related to DNS and DNSSEC.

ROY ARENS: I'm sorry, I can't quite hear you. Could you get a little closer?

HOSAM HASSAN: Actually, DNSSEC builds the trust relationship between the root and the other top-level domain, for example .com, and other domains, but if the case is that the hijacking happens on the resolver itself, if it's on the ISP or the end user, if the end user is using that its own DNS. I know the resolver contains addresses of the root servers, so if the hijacking happens on this stage and it points or routes the traffic to another root server, which is outside of that scope – so DNSSEC has nothing to do with this. Do you get my question?

ROY ARENS: I think I understand what you're asking. If you could end up as a resolver operator that's working for you uses a different route that's not DNSSEC signed…

HOSAM HASSAN: No-no. See, right now, as we've seen, the trust relationship is between the root, .com and under layer domains. Here is the resolver. He, the

hijacking happened on the resolver itself, on the ISP itself, before going out to the root.

ROY ARENS: if the question itself gets modified…

HOSAM HASSAN: On the resolver itself. To point to, IP address outside of the root servers, IP address, whatever, his server or another server, it would be out of the domain ecosystem. It will go out to another server that may contain another root server that way.

ROY ARENS: I think… Yes, go ahead Ray.

RUSS MUNDY: I'll repeat the question first in the way I think I understood it. I apologize beforehand if I've got it completely wrong, which is highly likely. The way I understand your question is, what happens if the root hints on the ISP resolver are compromised; compromised in a way that the IP addresses are changed to point to a completely different sort of servers, not the ICANN roots servers?

HOSAM HASSAN: Exactly. That's my question.

RUSS MUNDY: My answer to that is, if the root hints are compromised, but the resolver is still capable of doing validation, because it has the DNS key, if it has the DNS key it will still try to validate that information from the root. Since this is an alternative root, the information is obviously wrong and validation will fail. That means that the information will not go to the end user.

HOSAM HASSAN: No-no. Actually, the hacking happens on the resolver of the ISP, before DNSSEC is activated. DNSSEC is activated between the TLD, the root server and the under layer server, but when it's initiated the traffic from the ISP, it points to a database or a server out of DNS scope, another database, another ecosystem, another DNS database – at that stage, DNSSEC has nothing to do with that and the hijacker can point to another. In one click the IP address is 6.6.6.6 instead of all of this journey.

RUSS MUNDY: Let me point a little bit here. The resolver we're talking about is the one up at the top here – 10.1.1.253. That's where Joe User is sending his first query to, that's the ISPs. If any of the machinery of the DNS chain is in fact significantly compromised from software running properly on that machine, the results are completely unpredictable. They can do anything to change anything, if it's compromised.

You have to either have confidence that your ISP is working properly, or not. The point, also, of having reliance on a professional operation is

they should do it right. DNSSEC cannot guarantee your computer's security requirements, your physical requirements, your other protection requirements will be met, and they do have to be met in addition to DNSSEC, to be able to ride aligned with DNSSEC. Warren, did you want to add something?

WARREN KUMARI: Thanks. Warren Kumari, Google. Yes, if you don't trust you ISPs resolvers, or if your ISPs resolvers aren't trustworthy, they can lie to you. There are a number of people who've written software so you can run your own resolver locally on your computer. The Unbound software, for example.

HOSAM HASSAN: The end user can use this on the DNS or the ISP resolver.

WARREN KUMARI: Yes, and running something like Unbound locally on your machine, I think is a really good idea, because that way you can bypass trusting your ISP. You don't have to trust anybody other than your own computer.

RUSS MUNDY: In fact, if you look at the browsers I'm using in the illustration here, the browser itself is doing the DNSSEC validation. It happens to be what I have on my machine, and I'm sure most of these guys have something of the equivalent on their machines. It's not essential to rely on your ISP.

Most people would be able to do it more easily and more quickly relying on their ISP, and they're going to rely on their ISP for lots of things. So relying on them to do the secure and proper operation is one more thing, but it's not essential for the success of DNSSEC. Does that address what you were asking?

HOSAM HASSAN: It addresses the issue, but actually, to be honest with you, I didn't find the right answer until now, from my perspective of just how to secure the end user on the resolver's point of view, not from the DNSSEC life-cycle.

JULIE HEDLUND: Russ, I might suggest, just so we can move along, I think there will be a number of questions. Perhaps we should ask people to save their questions until the end of the presentation?

RUSS MUNDY: Let's go ahead and do that. Let's go onto the next slide. This won't take a whole lot more time. Now, in the upper part you'll see what's called the Bloodhound browser now, it's the name of it. It's a derivative from Firefox, Mozilla, and it has full DNSSEC capability, so it did all of the checking, and the hijack was going on. The hijacked information is not present on there.

This is your regular browser again and it is literally – if you look at the URL bar at the top – they were sent to the same place and they were

requesting the same information. You can see the non-DNSSEC browser did receive the hijack information pointing out that Steve Crocker finally admits that DNSSEC won't solve world hunger. It's intended in this case to be a very obvious and silly thing, it's something that's not real, but that was from the hijack. That's specifically content inserted from a DNS hijack.

Now, you look at web page and you say, "Oh, it's only that URL bar at the top." This is a picture of how many DNS queries it takes to fill your standard, commercial website, five years ago. That's what it takes today. To fill cnn.com it takes a huge number of queries. The important thing again is to get the zone data correct, and that's what DNSSEC is all about. The crypto's in the essential mechanism to make sure that happens, but it's the zone data itself that really is the crucial information that you want to get to the right place in the proper state, and be correct.

Here's another picture that simply shows what happens. I won't go through the data flow, but it's another query response like we saw earlier. Of course, these slides are on the website and if you want to refer to them later that's fine, or make use of them, perfectly fine also. This is your standard query.

Then the next thing is putting DNSSEC in. Depending on where you are in this whole chain of DNS, you saw an ISP here doing primarily recursive resolver. You saw the root, you saw .com, you saw .bigbank, and they all had different roles in the DNSSEC realm. Wherever your role is for doing

DNS related things, the general recommendation is that you should do the same type of things when you go to do DNSSEC.

So if you're doing it all in-house, because DNS is critical to whatever your mission or function is – if you're a TLD operator you're going to have a huge amount of DNSSEC expertise. You should do your DNSSEC probably in-house, with the same people because you have a very good, very high qualified, competent staff.

If you're an enterprise and you're outsourcing it to another activity, you should probably outsource it. Hopefully that same activity will be able to do DNSSEC but if they can't then you should look for another activity that would be able to do DNSSEC for you on an outsourced basis, unless you want to build DNS expertise in your own organization.

However you would do it, if you are an end user and want to use DNSSEC today, if you're a Mac user, which there's more and more of – and the reason we built it first for Mac is that it's the easiest platform to build for this sort of thing – you can get the Bloodhound browser right now, today, and you can run DNSSEC on your end machine as your default browser, and you can do DNSSEC right now, today, yourself, easily.

The zone data, DNSSEC is what gives you that assurance. It's what, in the final word, what goes into DNS by the authorized people to put it in for a zone, is what the DNSSEC tells you it's getting, to the place you're doing the validation. Whether it's your end machine or whether it's your ISP. In the United States, a very large cable provider called Comcast is doing DNSSEC validation on all of their resolvers for something like 18

million customers. Huge base, very large ISP, and they're doing the ISP-type of DNSSEC validation.

You can still do it on your end resolvers. I'm a Comcast customer and I do it on my end resolvers too. Oh yes, thank you Warren. Google… Anybody anywhere in the world, 8.8.8 and 8.8.4.4 are all doing DNSSEC validation. So you can use DNSSEC validation right now if you want to. The link between the validator and your end machine won't have any particular security, but you can use DNSSEC by using the Google validators too.

In putting DNSSEC in – this is a simplified illustration – but really all you need to do is get the data signed, and then you need to get it validated somewhere, whether it's on your machine or on your validating recursive server. This shows it on the recursive server, it can just as well be in the end machine. Again, the general principle… Yes, Olaf?

OLAF KOLKMAN: Just to put a number on that, Geoff Huston from APNIC did some research and looked at how many of the population that he sampled was protected by DNSSEC, and the size of that population. This is a global experiment. It's 8% of browsing Internet users are protected by DNSSEC. That's largely due to the Google infrastructure, but that's a significant amount of Internet users that are protected by DNSSEC from the customer side.

RUSS MUNDY: Great, thank you. I'd forgotten about those numbers. Yes. So wherever you are for doing your DNS, whatever your DNS operation may be, you should use the same general approach for doing DNSSEC and make sure – and this is an important aspect – whoever your providers of your software or hardware solution are, you need to make sure you ask for DNSSEC support. For many years there have been a lot of providers saying, "Where's the demand?" so people need to ask for it.

If you aren't, if it's not available now, this is also a time to maybe look at changing what some of the suppliers or whatever the source is. That is the set of information that's for the planned presentations. Now we're totally open for questions, comments. I do have one question I'd like to ask, folks. I had the slides that showed a hijack, and described the hijack, you saw the hijack in the skit.

Would people make a comment on whether or not you'd like to see it done for real? So we could set up a wireless network and hijack your computer, or hijack your computer? Would that be valuable to people? We're getting a yes. Okay. If this is something that's desired we'll look at doing it for the next ICANN meeting, as part of this, to show people this is real! Okay, does anybody have other comments on that? Yes?

AUDIENCE MEMBER: I have a question, not a comment. Is the floor open for questions now?

RUSS MUNDY: Sure, go right ahead.

AUDIENCE MEMBER: Okay. My question is more as an end user, more than being an ICANN Fellow. Talking about all the threats and safety issues around the DNS, I want to know, does the same thing apply to the applications? For most of the frequently used websites we use now they have applications, we don't go to the website. Do applications add one more layer, or is it more risky to use the applications?

RUSS MUNDY: Julie, could we go back a couple of slides? The browser that you saw is an instrumented browser that does DNSSEC. You do not have to have applications that are DNSSEC aware, but it's very useful if you do, to be able to see that content – the checkmark? – you do have to have DNSSEC aware applications. There is work going on for an application programming interface. There's been a fair bit of work on that over time and trying to get it through where applications can standardize it.

ROY ARENS: I interpreted your question slightly differently. Non-browser applications, anything basically, stuff on your iPhone, can use DNSSEC. DNSSEC is independent of the application, so if any application would use DNS, and your resolver or your ISP's resolver is DNSSEC able and capable, and stuff is switched on, all applications would benefit from it. I hope that answers your question.

OLAF KOLKMAN: May I put a different spin on this? Whenever you use the Internet, with everything you do in life, how your organize your life, essentially you use the DNS – whether it's your agenda, whether it's the news, whether it's sending mail, whether it's sending an IM, whether it's doing a telephone call over the Internet – everything in the back-end uses the DNS as a resource to get somewhere.

All these attack factors, all of these things that I just mentioned, have value. Hands are also an attack factor. If you turn on DNSSEC you introduce an extra layer of security for those applications that are not necessarily things you type in or use, but use the DNS in the background. That's the same answer in different words.

ROY ARENS: Thank you Olaf.

JULIE HEDLUND: We have a remote question.

SPEAKER: This is from Ade Bumbekimbo. The question is, how does a ccTLD deploy and implement DNSSEC? What is needed to make sure it works, and who are the parties that need to be involved?

ROY ARENS: Okay, well, I actually happened to work at a ccTLD, .uk. We have gone through this exercise, but what you've asked is a very broad question. If

I were to answer that it would be a very long session. Nothing here is new. A lot of ccTLDs have already deployed DNSSEC, there's an enormous amount of information out there. There's DNSSEC-2.org, from Russ's company.

Essentially, what you need to do is sign your zone, securely store your keys, you need to get your DNS record up to the root server, you need to do an enormous amount of testing, because when you sign stuff it basically means that you… You basically have other people authenticating it, so you'd better make sure it's correct.

So it's not rocket science anymore. Maybe ten years ago it was very hard to do all this, but nowadays there are a lot of tools out there, a lot of documents out there. Russ and his slides have many different links that contain all that information. Maybe Russ would like to elaborate a little bit?

RUSS MUNDY: I just wanted to point out some of the counting that's going on, approximately one-third of the TLDs are signed in the root zone now, and it continues to grow. Its rate of growth continues to increase. Earlier, as I talked about, however you do your DNS function today, if you're a TLD provider, an owner, and you have a registry partner that's an operator, you may already have a registry partner that's already capable of doing DNSSEC.

It may be no more complex than discussing with your registry partner if they currently do do DNSSEC, and have it turned on for your zone. Or it

may be, as Roy talked, starting from ground zero, without actually having it in, doing the planning and testing and making sure that it's right before you turn it operational. Over here? Go ahead.

AUDIENCE MEMBER: Has DNSSEC been used for email? 40% of the world's email is spam. I work for several companies as a consultant that send out hundreds of millions of emails, and the biggest problem is getting the legitimate emails cut by the spam filters. What's being done in the DNSSEC world that would help that situation? Thank you.

RUSS MUNDY: Well, I think it was Roy that mentioned this earlier, or maybe it was Olaf, but any applications that are running on a machine using DNSSEC, can take advantage of it. To this point there has been some implementations that are specifically pointed at SMTP and even IMAP; being able to see whether or not DNSSEC was used underneath it. But the most current activity that's going on is some work in the IETF that would make use of the DANE technology in conjunction with DNSSEC, that would help strengthen the SMTP to SMTP link. Roy?

ROY ARENS: For email there's… For complications between MTAs there's this thing called DKIM, DMARC, etc. These are protocols being invented, deployed, standardized, etc., that eventually can make use of things like DNSSEC or any kind of security mechanism, basically, that's based on a cryptographic principle. However – and I really like this quote; it's not

my quote, it's from someone else – bank robbers wear seatbelts too. DNSSEC is a seatbelt type of technology. It only solves small parts; the DNS parts.

If spammers use DKIM in such a way it eventually uses DNSSEC, then you haven't solved the spam problem this way. I hope this helps. Sir?

PAUL DONOHOE: Hi, Paul Donohoe from the UPU. We're the sponsor of .post. Hi Russ, we've met before in Italy on a DNSSEC conference. Thanks for your very simple explanation of a very complex topic. I have a couple of issues that I wanted to ask you about. The first is .post is entirely DNSSEC, so all the domain names in .post are DNSSEC signed. It's a challenge for our community, particularly as the gTLD we're really doing a lot of work in Africa and South America and Asia for domain names.

We find that there's a very slow adoption within the infrastructure within those areas. One of the challenges that we face, and one of the questions I have for you, is how do we get greater adoption within the infrastructure in these countries? What can we do to encourage the adoption? The second question is sort of related – how, as a user sitting in front of my computer, do I feel more comfortable with a website that I'm currently at? How do I tell that this is secured in general computer use?

This is one of the questions of many of users of .post domains. I tell them, "We're using DNSSEC, it's all signed, it's all secure," but how does the end user feel that comfort, feel that security, if they're using some of

this general software that's available every day? I think that's one of the challenges.

ROY ARENS: I'll try to answer that. Let me first repeat your question, and I'll make it a little more generic if you don't mind. The question is, how can we help the adoption of DNSSEC within the second-level domains, under, for instance .post or new gTLDs?

PAUL DONOHOE: Just to clarify that, I already have all of the domains signed, so it's how do the owners of those domains have access to DNSSEC security so that this signing can happen? I have the infrastructure in place in the registry already, with my registry operator, and I'm asking people to give me DS data. How do they get this DS data? They need to have that capability in their infrastructure already, yes?

RUSS MUNDY: Now, if you're asking about the names underneath of .post, to get those signed or the…? Okay. The registry operator is ready at .post, but whoever is operating the DNS at the next level, whether it be a country, which I would guess in .post's case would be very common. Or, another entity for, say, the settlement things or whatever – if there's not a policy mandate for is, people are reluctant because it is hard.

As Roy said earlier, there are a lot of tools, there are a lot of things available that are free, so it becomes a matter of knowing about them,

knowing where they are. If the people are the next label down have the expertise to do it, however they're operating their DNS today, if its being done as a service, an out-source for them, to another name server type of operator, then they need to find out if that name server operator is capable of doing DNSSEC.

If they're not, the big lever there is that they take their business to one that would. There are a number of name server operators that are trying to make their stronger business case because they do DNSSEC. Unless there's a policy stick it becomes a matter of helping lead them with the carrot rather than a stick.

ROY ARENS: There was a second question as well. Maybe on the topic of the first question, Sweden and the Netherlands, .se and .nl, they're basically the poster boys in terms of adoption of DNSSEC. What you see there, I think part of the reason of their success is they do a little marketing and give a discount if you register a domain and have DNSSEC deployed. That's one way of going. I'm not sure if that's the right or wrong way.

Your second question was how does the end user sitting behind a browser benefit from all of this DNSSEC stuff? Is there a signaling mechanism? How can a user see that it's behind a DNSSEC validating resolver? How can it feel secure? Currently in production there's no such thing. If you download a browser or use a browser standard with OSX or Windows, you won't see a thing. There's nothing there. If it's deployed it's mostly deployed at ISPs or geeks like us; TLDs etc.

But currently there's an effort underway to get a signaling mechanism, and Olaf Kolkman, right here, I've heard him say this afternoon something about a project he's doing together with a very large TLD, in order to at least get a standard about this signaling mechanism towards the user. I don't know if you can elaborate on this, Olaf?

OLAF KOLKMAN: Yes, I can. I've been interrupting a few times without introducing myself. My name is Olaf Kolkman, I work for NLnet Labs, and NLnet Labs has been working with guys like Russ in a small community for over a decade to look at DNSSEC and DNSSEC deployment and see where we can make a difference. With respect to what you just mentioned, what we're now interested in is specifically that last mile – how do you get DNSSEC to the user and how do you get it into the application?

The question that you had was basically a user interface issue. That's not where our specialty is, but also those applications will need to know whether DNSSEC is being used. Current APIs do not offer that capability, and there's work underway within IETF and a bunch of people who are trying to develop an API that would offer that capability to programmers. That might take a while to get there.

But I want to take a step back because I had a different approach to answer your question. I did some thinking about what it takes to do innovation of this type on a global scale. We have the same question with IPv6 – how do you innovate at the core infrastructure of the Internet? Now, if you look at innovation in general, there's work done by marketers like Everett in the '60s and they basically looked at what

are the things that people consider when making a choice for adaption; for adaption of an innovation?

Things like relative advantage, the complexity and simplicity of an innovation, the compatibility, the try-ability, and the observability of an innovation come into play. Observability of an innovation is just what you just talked about – how can a user see there is an advantage to using this? DNSSEC is technology that lives under the hood, so it has all of these issues that I just described; relative advantage – it's very hard to see, it's very hard to give that message.

But it is there, and it's all long-term advantages that we're talking about. It is all long-term security for a global infrastructure. It's not directly something you turn on for yourself, specifically at the site of the provisioning side. It is something whereby you protect the common good of the Internet and help to innovate.

What we've been thinking about collectively, I think we share that mindset, is how we can make sure that the try-ability, the complexity and the compatibility become lower, the barriers to that become lower. That's by doing free software and by creating tools. For instance, the tools to do the signing, which are available in many open-source projects. Indeed, another part of that relative advantage is creating incentives.

In the Netherlands there was basically a subsidy. If you sign your domains your registration fee becomes lower, and it's a small amount but for hosting companies that have 100,000 domains, that was a very interesting proposition.

So as a registry, at the registry level, those are the levers you can pull. Can you make sure that your community of users have easy access to the tools that are needed, and can you give them a financial incentive and a relative advantage? But that is all from a drive for public common good. I hope that gives a less technological perspective on the question.

PAUL DONOHOE: I totally agree with the infrastructure issue, and that's why .post has adopted a policy of 100% DNSSEC, because for securing the transactions that are going to be happening across the infrastructure, it's vital. I just want to come back to the observabilty issue, because once again it's the end user who's really going to drive adoption, because that's the business. We'll have a discussion offline about that because I think that's a really interesting thing that we should be focusing on.

RUSS ARENS: Right, and there's also… Our other session that we have, a full-day session on Wednesday, a lot more of the topics are talked about in a lot more depth. One of the important topics that I think will help, in terms of getting things to the user level, is the incorporation of the DANE technology, which we've mentioned a couple of times here. You can hear more about that in depth on Wednesday, or we can talk afterwards.

But that is also something that's very much moving… A way to move the information in front of the user so they can see. More questions? Back in the back – is this a chat room question? Go ahead.

SPEAKER: This is a question from Carlos Watson. What is the challenge by the ISP in order to promote adoption of DNSSEC into the resolver, that they manage is an issue to resolve, around there world of several hundred million of public DNS? Thank you.

RUSS ARENS: I didn't really catch the question. Did you Roy?

ROY ARENS: Would you mind repeating the question and maybe rephrasing it slightly?

SPEAKER: What is the challenge by the ISP in order to promote adoption of DNSSEC into the resolver, that they manage is an issue to resolve?

RUSS MUNDY: If I got the question right this time, I think the person was asking what it takes to get the resolver operators around the world to implement and begin using DNSSEC. Does that sound close to what the question was, I think? Yes? Okay. Well, one of the important things that is a big factor is that there is a demand. Everybody in this room can go back and look at what your ISP is doing. Whether it's your work-work ISP or your personal home ISP.

If they are not doing DNSSEC – and 8% is a pretty good adoption rate, from what we were seeing before, but that still leaves 92% that aren't. Go ask for it. That's one of the biggest drivers that there is, is for people to raise the issue and raise it in a way that they understand it. Demand is the best thing if you're talking about a competitive kind of environment, like most ISPs are. In other environments, come countries, it's still government controlled and still heavily government influenced.

So those that are able to work with the governments that may be involved in controlling this aspect should work to promote the ISP adoption. Warren?

WARREN KUMARI: What a number of people have done, like Comcast, is make somewhat of a sales pitch with this. There's a small cost to enabling DNSSEC on your ISP resolver. It's a little bit of additional overhead, it's a little bit of work, but it's not that much. What you can do is once you've enabled it, you can tell your users what a great job you're doing of aiding them with security. You can use this as a fairly good marketing or sales trick to tell your users that you care about them, you care about keeping their interactions on the web safe.

So ISPs can use the fact that they do DNSSEC as a sort of differentiating factor. I know that if I was trying to choose an ISP I would rather choose one who seems to care about security and wants to help me keep my banking stuff safe, than one that doesn't. I think using this as a sales tactic is a very good trick. Also, once a number of ISPs in the region do it, if you're not one of the ones doing it, and something happens to one

of your users, they could legitimately ask why you weren't doing everything you could have.

Were you doing your due diligence do keep them safe? So eventually, if you're not doing it you could be opening yourself to some legal risk in some jurisdictions. I don't know if that was helpful to hear?

RUSS MUNDY: Excellent, thanks. Yes, Joyce, go ahead.

JOYCE: I just wanted to make sure that I understand the whole thing. Maybe you can clarify it for me. In terms of the DNSSEC players, if the root zone is signed and the DNS server provider, the company that manages the DNS server, is also signed, so all the domains are signed, then who else has to also do their job in order to do that?

From what I heard, you said that the ISP… For example, at home I'm using a cable company to access the Internet, right? So, are they also responsible if they wanted to do that, they also have to do the DNSSEC things? I thought that once the name servers sign, it should be secure? It sounds like a lot of players had to chip in their work.

RUSS MUNDY: Let me see if I can clarify this for you Joyce. What you saw earlier in the sketch was, the first part was doing it without DNSSEC, asking the query and getting the answer. Our Mr. ISP, Jacques Latour, was the one going and doing all the questions to the various name servers. After that, then

Dr. Evil came in and took it from the very last place, but he could have taken it anywhere along the way.

Now, when you sign everything, if you don't have a validating name server resolver, like Mr. ISP in our skit, then the user wouldn't even know whether or not the data had been signed, let alone having the information to do the validation.

So there's an important role for the resolver or ISP operator – or the end user – to actually do validation of the queries. So first you have to get the data in and get is signed. That part you had absolutely right. But unless the validation of the correctness of that data is done at the time of the query, you do not get the advantage of DNSSEC. Does that difference make sense?

JOYCE: Yes, but when a domain is signed, a DNS is signed, you're talking about APIs, and this website is developing the API also – do they also have to… The API, the program they want, does that also have to be signed, in addition to the domain name that's been signed?

RUSS MUNDY: No, the applications do not have to know anything at all about DNSSEC, as long as their DNS resolver is doing the DNSSEC validation. Whether the resolver is on the user's machine or on the ISP. Applications don't have to… It's better if they do, but they don't have to know anything about it. Okay, any other questions? I think we're close to the end of our time. One in the back? Yes, please, do you have a mic?

CRAIG NESTY: Craig Nesty from Dominica. I noticed in your skit, at the last step when the hijacking was taking place, the hijacker did not run DNSSEC so the resolver basically caught that. My question is, what happens if you have a host server who does not run DNSSEC and everybody else runs DNSSEC? Is DNSSEC backward compatible with just normal DNS, or would it reject that website completely?

RUSS MUNDY: The design of DNSSEC accommodates things that are not signed, as well as those that are signed. If in getting the answers to the DNS query it comes to a point in the tree where there's not DNS signatures present, that's clearly identifiable within the DNSSEC protocol and the way the DNSSEC resolution works. It recognizes it as not signed and it will just work, and the information will get resolved. So it won't be rejected if it's not supposed to be signed. Yes?

AUDIENCE MEMBER: I am the developer of NIC Argentina. I could make an application that detects the sign in every step of the chain. I'd make a cell phone application and I want to make it… I want a program to tell me that the connection is DNSSEC. Could I detect it?

RUSS MUNDY: Well, there are implementations that run on cell phones now. Some of the things in our toolkit run on Android and Maemo, which are

cellphone OS's, so it's certainly very doable, and if the data is signed it will be validated. It's a matter of examining what your code base is, what the application is that you want to use, and then proceeding with reading the specs and writing the code.

So yes, it's very doable, and we'd love to see more people doing it. Please go for it. If you need help this is a very incredible helpful community. Our tool site gets a fair amount of traffic. NLnet Labs gets a number of question about both what they build and about help for people that are doing more in the DNSSEC world. So please proceed. It's achievable – it's not necessarily that easy, but if you're family with a platform and a software, go for it.

JULIE HEDLUND: So, Russ, we are a couple of minutes past 6:30. We could take maybe one more question?

SPEAKER: There has been a gentleman behind the pillar over here trying to get your attention, if we could go over here.

AUDIENCE MEMBER: My question is, is there any relationship between DNSSEC and SSL – secure socket layer?

RUSS MUNDY: They are actually two independent protocols. They're complementary to each other, if you will, in that one works just fine with the other. You'll get stronger security if you use both, in terms of your total security, but they are separate but they can be run at the same time, on the same machine.

I suspect some of these guys sitting over here are doing it. My laptop's not on so I'm not, but I do it all the time. It's very good. Multiple security at different protocol layers is a good and useful thing to do. Did that get your question, sir?

AUDIENCE MEMBER: Thanks. Yes, sure.

RUSS MUNDY: Okay. Thank you. I guess that's it. Well, thank you everybody. This has been a really good session and we try to do them each time. if you're at the next meeting you're certainly welcome to come and join in again. Thank you.

JULIE HEDLUND: Thank you.

[END OF TRANSCRIPT]