LONDON – DNSSEC for Everybody: A Beginner's Guide
Monday, June 23, 2014 – 17:00 to 18:30
ICANN – London, England

UNIDENTIFIED MALE: It is Monday, June 23rd. This is the Thames room, and this is DNSSEC for Everybody: A Beginner’s Guide. 5:00.

JULIE HEDLUND: Welcome, everyone, to the DNSSEC for Everybody: A Beginner’s Guide session. My name is Julie Hedlund. I’m with ICANN staff. We’ll start in just a couple of minutes. There are seats at the back. There are some seats at the table. Please do take any available seat. I’m sorry the room is little small. We might have to be pushed in a little bit. I apologize for that. Please do try to find a seat and we’ll start very soon.

UNIDENTIFIED MALE: Thank you all for coming. We will get going in just a minute. We need another person for our skit, who should be here shortly.

ROY ARENDS: Hi, everyone. My name is Roy Arends. Thank you for coming. We will start shortly, but we were waiting for—

UNIDENTIFIED MALE: We got him.

Note: The following is the output resulting from transcribing an audio file into a word/text document. Although the transcription is largely accurate, in some cases may be incomplete or inaccurate due to inaudible passages and grammatical corrections. It is posted as an aid to the original audio file, but should not be treated as an authoritative record.

ROY ARENDS: We're not waiting anymore.

DAN YORK: Yes, we can start. Good afternoon, everyone. Thank you for coming here, finding your way through the Hilton London Metropole to get to this room. This is part of the journey here. My name is Dan York. I'm here with the Internet Society, but I'm part of a group of folks who for the last several years have been going and providing these introductions to DNSSEC.

First question: how many of you have done anything with DNSSEC? Okay. All right. The rest of you just looking to understand more of what it's all about, eh? All right, good. I see some waves here.

In this next little bit, we've got some slides to talk a little bit about what DNSSEC is all about. We're also going to do a little skit up here where you'll see a bunch of us coming up here and talking about DNS and how it relates to DNS versus DNSSEC.

I'll also put in a plug that if you're interested in more about DNSSECC, we have a full day workshop happening on this Wednesday, starting at 8:30 in what room, Julie? It's the Hilton 1-6, which is – I'm not actually sure where that is, but it's somewhere in here. But Hilton 1-6. It starts at 8:30 in the morning, goes to until 2:45. You can come in and out. The agenda's posted. We've got a lot of deep technical sessions where we're diving into a bit more about DNSSEC. We have some presentations around DNSSEC in Europe – part of pieces around there. We've got some other sessions diving into some of the crypto modules. We've got a bunch of demos in the later part talking about this thing called DANE and some other pieces. It's a great workshop. We encourage you to come and check it out.

With that, we'll start to begin here. The schedule of what we're looking at – and I would also mention, too, we do have people coming in remotely through the Adobe Connect room, so we will be trying to use the microphone to help people hear out there, and if you ask questions, we'll want to get the microphone to you as well.

So I'm going to start with a little bit of a fun story about kind of painting the picture of why DNSSEC matters in here. Then – oh this is the wrong person. It's not Warren. Hey, Roy, you almost got out of here. Okay.

Roy is going to talk about the basics of DNSSEC and DNS and DNSSEC – how the parts are. In the midst of this, we're going to do this little skit I referenced, where we talk about DNS, DNSSEC and how that works.

Then Russ Mundy will be here to go into a little bit more detail about some of the pieces and the ways this works with a couple of posts. At the end, I'm going to come back up and we've got a session, an interactive part, where we would encourage you all to ask questions. The last time we did this at the last event, we had a lot of discussion, a lot of good questions. We're here to help you to talk about this, and we'd love to help you work with that.

This is the DNSSEC for Everybody. Come on in if you'd like. There is space around the back and around the side.

To begin with, I want to jump back a little bit. We're going to talk about the beginning of DNSSEC, if we will. We've got our little story here. This is Ogwina. She lives in a cage on the edge of the Grand Canyon. This is

Og. He lives on the other side of the Grand Canyon. It's a long way down, but they want to be able to communicate and talk back and forth. They'd like to have some way to go and do that.

On one of their visits, they notice that there's smoke coming from the fire. They said, "Oh, hey, we could send each other messages." Soon they're chatting regularly using smoke signals from one side to the other. Great form of communication here. They're able to talk without having to go down there.

One day, mischievous Caveman Kaminsky comes in next door, and he starts sending smoke signals, too, and so suddenly Ogwina is really confused. She doesn't know which smoke is the one that she should be paying attention to. "Which one is it? I don't know. There's multiple smoke up there. One of them is the one I want to look at. What is it?"

So she sets off to try to get over to the other side, and she and Og consult the wise village elders. One of them, Caveman Diffy, has an idea. He goes up and he goes back into Og's cave, where at the back he finds a strangely colored sand that's only there in that particular cave. He goes and he throws some of this in the fire, and all of a sudden that smoke turns into blue. Just little bit of this each time, and all of a sudden now Ogwina and Og can chat happily because they know that all she has to do is watch for the blue smoke.

Believe it or not, this is really what DNSSEC is trying to do is to provide that blue smoke, if you will, but the way to identify clearly that the information you're getting is truly coming from the source where you wanted it to be. With that, I'm going to turn it to Roy to jump a little bit more into it.

ROY ARENDS: Thank you. I’m going to set up for that. Yes, please. Let’s go to the next slide. Thank you.

So in order to understand DNSSEC, you kind of need to understand how DNS works. I’m going to give you a high-level concept of DNS, not because I don’t think you wouldn’t understand it if I go into the technical level too deeply. I’m just a little bit afraid you’ll all for asleep.

The DNS is basically a tree – and inverse tree, if you will – with the roots at the top and all the other labels (the domain names consist of labels) are hanging off that root. For instance, you can have BigBank.com. BigBank.com actually starts at the root. That’s the label you don’t see, and then com, and then com delegates the information to BigBank.com. Next slide, please.

How it really works in the real world? There is a resolver. Your ISP accesses that resolver, and on your laptop, you actually talk directly to that ISP. The resolver then first goes to the root and asks the root for, “Where is www.BigBank.com?” Then the resolver knows where the root is, goes down, traverses that tree, and eventually ends up at BigBank.com.

In order to show you what really is going on, we would like to give you a play with a few of my friends. We’ve of course never done this before. Shall I do the introductions really quick, or…?

UNIDENTIFIED MALE: Sure.

ROY ARENDS: Okay. We’re all going to play a part in this. I am going to be a root server the moment I have this T-shirt on. We have Dan York, who’s going to be .com. We have Julie, who’s going to be BigBank.com. We have Cass. She’s going to be the ISP resolver that walks around a lot to get all the information. Then we have Joe User, who actually talks to the ISP resolver.

DAN YORK: Okay, thank you very much. Okay, so we’re going to take these very intelligent DNS experts and turn them into actors. The first thing we’re going to do is show you a DNS transaction as it would occur normally. Then we’ll step you through four acts of this play altogether. This is the first act. This is Joe User going to the bank to do his online banking.

JOE USER: So I’ll start off.

DAN YORK: Thank you.

JOE USER: I open up my laptop. Have to do some banking and pay some bills. I type in www.BigBank.com. Ms. ISP?

CATH GOULDING: Hi, Joe. Let me just check. No, I don’t know what that is. Hang on. I’ll go and ask Root. Hey, Root, do you know what www.BigBank.com is?

ROY ARENDS: No, I don’t. But I do have the information where Com is. Com is at the address 1.1.1.1

CATH GOULDING: Thank you. Hey, .com, do you know the address for www.BigBank.com?

DAN YORK: No, I don’t. But I do know who knows the domain names for BigBank.com. They’re at 2.2.2.2

CATH GOULDING: Thank you. Hi, BigBank. Do you know what the address is for www.BigBank.com?

JULIE HEDLUND: Well it turns out that I do. Thank you for asking. The address is 2.2.2.3. There you go.

CATH GOULDING: Thank you very much. Here you go, Joe. That’s what you need.

JOE USER: Thank you very much, ISP. Now my laptop can run off to this address – 2.2.2.3 – and that is BigBank, and I can happily pay all my bills at the bank.

So that is a normal transaction. I guess one thing to note there, me as a user, I didn't do anything other ask my ISP for the address and the servers – authoritative servers and the ISP recursive server – took care of all that for us.

UNIDENTIFIED MALE: [inaudible]

JOE USER: Okay. Now we're going to demonstrate a man in the middle of attack. This is really why DNSSEC was created. It was to present such attack. Again, same set up. We're going to demonstrate a man in the middle of attack at the speed of light.

Okay, more banking to do. A lot of bills. So go to my laptop, type in www.BigBank.com – Ms. ISP?

CATH GOULDING: Hi, Joe. No, I don't have it. Let me go and ask Root. Hi, Root, do you know what www.BigBank.com is?

ROY ARENDS: No, I don't. I do know where Com is. Com resides at 1.1.1.1

CATH GOULDING: Thank you. Hey, Com, do you know what the address is of www.BigBank.com?

DAN YORK: No I don’t, but I know that you can get answers for BigBank.com at 2.2.2.2

CATH GOULDING: Thank you. Hey, BigBank.com, could you tell me what the address is of www.BigBank.com, please?

UNIDENTIFIED MALE: I most certainly could. The address you’re after is 6.6.6.6

CATH GOULDING: Thank you very much. Here you go, Joe. The address you’re looking for is 6.6.6.6.

JOE USER: Thank you Ms. ISP, a great job as usual. You got that address for me. Now we can go off and do my banking online. I’ll be happy.

So what would happen, instead now of actually going back to going to BigBank, my computer got the address from the ISP is going to go to Dr. Evil here and they’re going to clone the bank’s website and all my money’s gone.

So along comes DNSSEC to save the day. Before we actually do DNSSEC, we have to form what’s called a chain of trust. These servers basically don't really talk to each other than know each other's presence. There's no way that, when the request goes in, whoever responds back the quickest – Dr. Evil – that's the address they take.

So there has to be some way of actually identifying each other, so they're going to create a chain of trust. It will start with the root.

ROY ARENDS: Hi, Com. I'm Root. Do you have some information I can trust?

DAN YORK: I do – whoops [inaudible]. Here's my signature, or here's my – yeah.

ROY ARENDS: Thank you. Thank you. Also what I need to do, I need the ISP to trust me. So, ISP, you can trust me. I'm the root. You already trust me as the root as all DNS, so you can trust me at the start of the chain of trust. There you go.

DAN YORK: BigBank, hi. Do you have something for me so I can trust you?

JULIE HEDLUND: Well, I do, and here you go.

DAN YORK: And then BigBank would also sign themselves, right? You have that on there, yes? There we go.

ROY ARENDS: Joe?

JOE USER: Okay. So now the servers have shared secrets with each other. So they know each other. They can identify each other and trust each other. That was really the key of DNSSEC.

Going back to the Ogwina, that's the blue smoke that's now been exchanged. They actually know something about each other that's special.

Let's do the same transaction again. More banking. And we'll do another man in the middle of attack.

Okay. So off we go again. More bills to pay. Go to my laptop. Tap, tap, tap. www.BigBank.com. Ms. ISP?

CATH GOULDING: Hi. Thank you. No, I still don't know that. I'm going to go and ask the Root. Hey, Root, do you know what www.BigBank.com is?

ROY ARENDS: No, I don't, actually. Com resides at 1.1.1.1. Here's my signature. We can shake on that. You can trust that information.

CATH GOULDING: That's cool. Thank you.

ROY ARENDS: I've got a badge.

CATH GOULDING: We've got badges. Hi, Com. Do you know where www.BigBank.com is?

DAN YORK: No, I don't. But I know you can get answers at BigBank.com at 2.2.2.2, and here's my signature to show that I know what I'm talking about.

CATH GOULDING: Yup, that's cool again. Thank you. Hi, BigBank. Do you know what www.BigBank.com is?

UNIDENTIFIED MALE: I certainly do. The address you're looking for is 6.6.6.6

CATH GOULDING: Hang on a minute. Where's your star? I don't think I trust that. No, I'm not going to trust that one.

I'll ask again – or don't ask again, do I? What is www.BigBank.com?

JULIE HEDLUND: I have the answer, and you can trust it. It's 2.2.2.3, and you see there's a big star and a nice handshake with it.

CATH GOULDING: Thank you. Here you go. You can trust that response.

JOE USER: Great. Thank you very much, Ms. ISP. So now I go off and do the banking and it's nice and secure. Dr. Evil is all frustrated and he's got get a new line of work, probably send e-mails about money I have coming to me from some bank account.

Again, the thing to notice here as me as Joe User, I didn't have to do anything. Nothing at all was different from my perspective. Typed everything in. Got the result back. Everything's happened at the server level, so on the infrastructure side of things. So at the ISP and the registries and in the hosting company at the end.

So that's it from the Famous Players Actors Guild. And Dr. Evil.

ROY ARENDS: Thank you. To continue this, I'm going to reinforce basically what actually happened with what just happened. Just to relate to the initial slides that you saw, you saw Cass the resolver chatting with Og the server. Actually, Ogwina talks to many different – Ogwina's a very modern woman – talks to many different Ogs. All of these Ogs are basically servers in this session. Next slide, please.

We saw Dr. Evil try to get involved. So Cass didn't know who the real servers were, so she just has to accept the information that came in. Next slide.

With DNSSEC, Cass, the resolver, or Ogwina, the cavewoman, can actually verify that she's talking to the correct server. Next slide, please.

Back to DNS, without DNSSEC there is no security in DNS. It is almost as old as the Internet is. DNS was invented in 1983. Currently, it makes 31 years? Is that correct. So yeah, 31 years old. Names can be easily spoofed, and caches can be easily poisoned. This was already known for a long time, but Kaminsky has shown actually how easy and how trivial this can be done. Next slide, please.

Yeah, I think we talked about this – thank you. So DNSSEC is the solution for that, and DNSSEC is the only solution for this. This is how DNSSEC works. DNSSEC uses something like cryptographic signatures – or digital signatures, if you will. Just like addresses that you store in the DNS for your websites, or for instance, MX records that you store for mail, you can store cryptographic keys in the DNS because a key is nothing else but a bunch of bits.

Now, it is the public key, naturally, that you store in the DNS, and the private key that you keep to yourself. There's a relationship between the private key and the public key. It's a mathematical relationship. Only those who have the private key can generate a signature, and those who have the public key that's related to the private key can verify the signature. So it is the public key that is stored in DNS.

And you know what? Signatures is binary information as well. Again, just a bunch of bits, so you can store that in the DNS as well, just like address records and MX records. Now we have keys and signatures stored in the DNS.

Of course, you cannot trust that completely because there still needs to be a link between com and BigBank.com, and this is actually nothing else than a short form of the DNS key. For BigBank.com, it's stored at a com level. Com signs all their information, and therefore you can trust the DNS key from Example.com because it's been signed by com. Next please. Thank you.

Back to DNS, there's a nice analogy between DNSSEC and DNS. DNS traverses from the root down. It goes to the root first, then to com, and then to BigBank.com. It's actually happening the same with DNSSEC, but not resolving, but a chain of trust.

So the ISP, the resolver, just says it knows where the root servers are. It also knows what the root key is. It has been in one point in time a secure exchange of information so the ISP can trust this root key.

So now, since the ISP can trust this root key, it can now traverse through com, it can traverse through Example.com , and builds this chain of trust, similar to, for instance, I trust Cass, Cass trusts them, and therefore I can trust them. Next slide, please.

Therefore, we can weed out the false information about BigBank.com because they are not able to falsify the DNS key. Only if the fake or Dr. Evil or Kaminsky is able to change at com level the key for Example.com, then there is some danger. But that shouldn't happen. So the fake BigBank.com can't really falsify the signature. This is what [Krypter] is all about. Next, please.

Perfect. That was my bit. I'm now going to hand over to Russ Mundy. Russ?

RUSS MUNDY: Thank you, Roy.

ROY ARENDS: You’re welcome.

RUSS MUNDY: So we have a few examples that we’d like to talk about here today. I’m Russ Mundy. Normally, at this point in the presentation, you’ve seen me in the skit because I’m usually BigBank.com, but Julie was kind enough to be BigBank.com for me today because I did have some other obligations that kept me from arriving on time, and I apologize for that. Next slide, please.

What is it that people need to worry about when it comes to DNS and why should you be concerned? That is, DNS in and of itself is important and critical to the operation of the Internet, but nobody does DNS just for DNS sake. You do DNS because you want to use an application, whether it’s Skype or e-mail or jabber or whatever it is you want to use. But you first have to use DNS on almost every application on the Internet. So if you don’t get the DNS right, then your applications aren’t going to work. That’s really the fundamental issue with getting DNS right. Next – yeah.

That then leads to the question, “Why do attackers go after it?” just like you saw in the skit. They weren’t after DNS to do bad things to DNS. They were after DNS to get at the banking application, or the jabber, or to steal your e-mail, or be a person in the middle and grab all your e-mail and deliver it on. They see all your e-mail and the recipient gets it, but you never know it as the user of an application.

So one of the things that has occurred over time is there are hijacking tools for DNS that are out there in the open source that are just simply available. All you have to do is go look and you can find them, and people do, and people do make use of them.

In fact, I have not found this in the last couple of years, but at one point, I found some coursework that was being taught on an online course that one of their assignments was to actually write software that did a DNS hijack. So it brings into my mind the question of, “Was this really an ethical university that was teaching this?” But yeah, apparently it was legit. So there’s a lot of information out there for how to do it.

How do you know that you’re reaching the right place? Like Roy just said, it’s cryptographically based, which really goes back to mathematics and the way that mathematics and the cryptography work is that the entity in DNS that’s doing the validation of the information that they get from the DNS can prove with an extremely high degree of certainty that it came from the right place, and that the answer was not modified in route. So it’s really those two things that DNSSEC does for you.

Now, if you ever happen to encountered a DNS hijacker that’s serious, they can actually create a website that looks like the one you were going to, which was kind of the example we were trying to show in the skit, because if a human being was going through their web browser and got to bank and the website didn’t look the same that it did last time, well, they’re not going to put all their information in. But, in fact, just as you can steal DNS, you can also replicate websites. Next, Dan.

The numbers on this picture illustrate the sequence in which this occurs. When the Joe User sends out his query, before he can actually get the website of this bank, he first has to do the DNS query, and that goes to the recursive resolver, and the recursive resolver goes to the authoritative server, the authoritative server gives the answer, goes back to the recursive, goes back to the recursive, and then finally the recursive resolver has all this information from DNS and then gives the answer back to Joe User, who, after all this happens, can't get to his bank.

So these are things that people just don't even see because it happens so fast. It's behind the scenes. This is an abbreviated example of the packets that flow. It's not at all unusual to have as many as 15 DNS queries occur before you actually get to a website. Next.

So when you get to a website that's set up to show that you're using DNSSEC – and this is one that my group provides for some of our tools – it actually illustrates that, whether you're using DNSSEC or whether you're not – but that's something that we've done just simply to make it visible. When you are using DNSSEC, you get the check. When you don't, you get this. That's an informal thing.

But what actually occurs is Dr. Evil Hijacker tries to get it, and the answer is simply thrown away because it fails the cryptographic validation.

When you look at the number of queries that it takes to fill any website that's a commercial-type website today, it is shockingly large because almost all these websites pull things from all over the world.

This is CNN about four years ago. It took about 75 to 85 queries to actually fill that page. We've done a check like that about a year ago and it took 120 DNS queries to fill a page. Next.

That's what the picture looks like more currently. The thing is, any one of those queries can be stolen, and you can have content substituted on you and you don't even know it.

The important point that a lot of people forget when you're talking about DNSSEC is, yes, it's cryptographic. It's crypto-based. There's keys involved. But why are you doing this? The reason you are doing this is to protect the DNS zone content data itself, whether it's the address of the webs server, or it's the mail server. Whatever you're trying to do to actually use the application, it's the DNS zone data.

So the lesson is you should protect your DNS zone data, just the same amount, no more security, no less security, than what you protect your cryptographic mechanisms you're using.

A lot of people think it's the crypto that gives you the security when you put the information in the DNS. That's not where the crypto gives you the security. Where the crypto gives you the security is Joe User down here in the corner where our ISP cat could tell that they made a mistake and somebody was trying to do a hijack. So getting the data in, you want to protect the data as much as you do the keys.

Another picture shows getting the data in. Next. Then when you actually put it in, you need to put DNSSEC to make it work in multiple places. You need to have your zone signed. You need to have, when you get the data out, you need to be able to validate the data as it comes out.

So if you’re a large company and you’re running around DNS for your own DNS services, for your authoritative servers, for your recursive servers, then you probably want to do all of the DNSSEC things yourself. If you’re using another approach, such as having a service provider do your DNS capabilities for you, you want to probably have that service provider also do your DNSSEC capabilities for you because if you don’t have the in-house expertise in your organization to do DNS, you probably don’t want to be doing DNSSEC.

For the things that are very much names-centric/DNS-centric kinds of businesses, like a registries services operation, they usually have a very, very strong in-house DNS expertise that they can apply also for doing DNSSEC.

If you’re an enterprise that operates a lot of what they do internally but may not want to do your external-facing DNS, then you may want to have that same company do your external-facing DNSSEC as you do your current DNS services.

For instance, Parsons.com is DNSSEC-signed. That is actually operated not by Parsons.com, but by an external service. The internal DNS is operated by Parsons the organization. So you have a mixed set of the organization that I’m with of ways of doing your DNS and DNS operations. So your implementation can be broadly different depending on where you are. You need to look at what your current situation is and what makes sense for putting your DNSSEC functions in place.

The other point that I wanted to get back to was the DNSSEC is not going to prevent people from attacking you. If they want to attack you, they’re going to attack you. What it will do is it will prevent the users of your DNS system from getting this bad data that comes from an attack. They will be able to tell if they've been attacked, either because the signature didn't validate or you had some other indication from the DNSSEC information that prevented that attack from succeeding. Again, biggest point is the DNS zone data is what's protected.

This is the simplified picture of your zone data. It goes in. Your authoritative server serves it. Here's Joe User over here. The simplest illustration of what it takes to do DNSSEC: your zone gets signed – okay, so you have signed DNS information for your zone – and your validator (in this case in the skit, our friendly local ISP) does the validation. And you saw in the skit how without the validation the wrong information comes back. With the validation, you do not return the invalid information. You can detect that. Then the valid information can get returned.

The approach for doing the various pieces associated with DNS, likes say varies by organization to organization. Over time, we've tried to give some illustrations of how one might go about doing it, and what we found is it's better just to leave it to high level, let people ask questions if they have specific questions about what they're doing. So we do have a question and answer session afterwards.

I think at that, it's time for that session, which I'll give it back to Dan.

DAN YORK: Thank you all for paying attention to what we've been doing here. I'm told the score is still 0-0 for people still concerned about that. You all should have a handout floating around. If you didn't get one, there's probably a few floating around. It's also on the website. If you go to the page for this, you can download this handout. We have some information on the back about some resources you can all have to learn more.

I'll make a couple little quick points before we go into question and answer. All these transactions – and Russ showed that picture of when you go to a website, it might be a couple hundred DNS queries to build parts of websites – all of that's happening like this. We're talking milliseconds. We're talking microseconds in some cases. There's just all these queries are happening. So all of this is going on at that insane speed, and so the validation fits in there as well. When you deal with validating servers, all that's happening right then, right at the wire.

Another point we talked about in our skit, we had Cath being the ISP, and the ISP, for most people, is probably where the validation happens. If you live in Sweden or the Czech Republic or the Netherlands, the ISP that you're using probably is doing DNSSEC validation right now for you. You've been having it right now all the time. If you're in North America and you use Comcast, they turned it on last year for their 18 million customers all across the country. If you use Google's public DNS as your DNS servers, Google does DNSSEC validation by default all the time. So it can happen at the ISP or public level. If you're interested, you could also turn it on on your own network. You could do it in your home network. You can do it with whatever you use for a DNS resolver if you want to experiment with this.

There are also software you can get to turn it on on your local laptop. You can run a local DNSSEC-validating resolver on your laptop or local device to be able to do this. So if you want to experiment and see what happens there, you can find out more.

If you look at the resources, we've got some here. I work at the Internet Society on a program called the Deploy360 Program, where we provide deployment information for DNSSEC IPv6 and some other technologies. We have a whole series of tutorials up there for people who are interested in learning more about this.

You'll also see on here we have a DNSSEC-Tools.org, and Russ referenced this. His group has some tools up there which have a number of different pieces you can use to do this, including a browser called Bloodhound, which you can install which does all of this DNSSEC validation if you'd like to experiment with that. There's some other references on here as well to some more points.

One other piece is, when we showed you the skit, each time the Joe User asked a question, Cath as the ISP went off and talked to the root, and then went to .com, and then went to BigBank.com to get the answer.

How efficient would it really be if she did that each and every time, right? That would be a lot of queries if she had to go to the root every time. What happens is – and this is where it gets dangerous – is that she caches the result. So once she knows from the root where the .com name server is, she files that away for future knowledge for a certain time to live, a certain amount of time. When .com tells her where BigBank.com is, she files that away in her knowledge. And when BigBank.com tells her where www is, she files that away for a certain amount of time. So the next time that Joe User wants to get to that banking website, Cath as the ISP is able to say, “Oh, hey, I’ve got that. It’s in my cache. I’ve got that information. It hasn’t expired, so here it is, Joe. Here it is, [inaudible].”

So she doesn’t have to do that whole chain every time, only when her information expires, which is where part of this whole – we call it this cache poisoning attack – where part of the deadly part is that Dr. Evil can insert his answer because in DNS, speed wins. Whoever gets the fastest answer back to the resolver is the one that the resolver takes. Until you get into DNSSEC, everything else is just speed wins. Dr. Evil gets there quickly, boom, she’s got that. She holds onto it and she gives it out to Joe User for whatever period of time is on that. So that’s the poisoning that happens in there. There’s this caching that goes on.

So that’s a bit of our story, our introduction here. Now we’ve got about 45 minutes that we’ve got remaining in the session, so we’d like to throw it open for you for questions, and the Dutch may tell us we shouldn’t take questions, but sorry, Roy, we’re going to take a few.

Yes, a question back there, and if you could turn on that microphone if they work.

UNIDENTIFIED MALE: Okay. Hi, everyone. I’m [inaudible] from Botswana. I kind of want to understand a few concepts with regards to DNSSEC relating to the man in the middle who acts as an intruder to the network. Is there any relationship that exists between the intrusion detection system and the firewalls as tools that we usually see implemented within the network, but exists between the DNSSEC and them?

DAN YORK: So there’s – do you want to say it, Russ?

RUDD MUNDY: Sure.

DAN YORK: Okay. I was going to say, Russ and Roy and Cath, feel free to jump in if you wish to on any these.

RUSS MUNDY: So the thing that are looked for with intrusion detection system and firewall systems have different characteristics than what occurs with the DNS-based attack – a DNS hijacker, a man in the middle type of attack – because as Dan said just a little bit ago, without DNSSEC, the very first answer that gets back to the DNS resolver that asked the question, that resolver will take the answer. It just simply has no way to know it’s coming from an illegitimate or an inappropriate source, because from a protocol perspective, which is what you look at in a firewall and intrusion detection system, as well as heuristic characteristics, there’s nothing unusual. It is just an ordinary DNS answer, so it will let it pass through these other prevention devices. So only the DNSSEC cryptographic mechanisms can really detect that.

DAN YORK: Now, having said that, there is one little thing is that firewalls and other middle boxes can prevent DNSSEC from working. Wes Hardaker over here actually has a draft in the ITF around called DNS Roadbloack Avoidance which points out that some of these middle boxes in the firewalls and other devices in there can do things that wind up blocking packets or blocking certain types of things going through, or rewriting DNS queries, doing different things. So there is some danger. You need to make sure that your firewalls and other devices in their – your NAT box and other things – allow DNSSEC to work through them and have the queries come in.

Otherwise, what Russ is said is right. DNSSEC is a different layer in defense. Did you have another question? Do you have another question?

UNIDENTIFIED MALE: No, that's all. That's all.

DAN YORK: Okay. Thank you. I saw this gentleman here.

ANUPAM AGRAWAL: Hello there. This is Anupam Agrwal from ISSOC Kolkata ALS. Okay. So you said that some of the ISPs enable DNSSEC. So it is it really expensive what it takes for ISPs to enable it?

DAN YORK: What does it take for ISPs to enable it? About one line in a configuration file to uncheck something. Seriously, in many cases, it's that easy. Most of the recursive resolvers that are out there now allow you to go and enable DNSSEC validation simply by a simple configuration factors.

One of the concerns that some of the ISPs have is more of the case of – well, let me give you the classic example. What was NASA – a year and half ago? Two years ago?

UNIDENTIFIED MALE: It was almost three now.

DAN YORK: All right. Well, okay. So when Comcast first went and enabled DNSSEC validation, NASA's website was signed and what happened was somebody at NASA forgot to resign it, because signatures expire, and so all of a sudden what was happening was people who were on Comcast network were not able to get to NASA's website, but yet if they pulled up their mobile phone, they could get to NASA's website.

Now, this happened, unfortunately for Comcast, it happened on the same day that there was the SOPA PIPA blackout that happened across all the websites that people were doing. They were all doing this thing. So all of a sudden, everybody thought that Comcast was like conspiring to block out NASA and all sorts of things. There was this big whole blowup on social media.

But anyway, but the net of it was, some ISPs got a little gun shy about not wanting to turn on validation because they were afraid that people couldn't get to some websites. But from a practical level, the actual increase to their own capacity isn't a huge amount. It's not a huge expenditure of additional machines or additional processing capability, at least that I've heard from any ISPs.

Do you guys have any feedback? Every ISPs I've had has said the validation component was relatively trivial, so it's more of an education about what's going on. And in some cases, it's an education of its customers, that we are now providing you a more secure experience, which means that if you can't get to a website it may be because there's some problem there and they're preventing you from getting there.

ANUPAM AGRAWAL: [inaudible]

DAN YORK: Yeah. Awareness is more of it. Yes.

UNIDENTIFIED MALE: Hi. [inaudible] from Fiji. I'm trying to get my head around how the trust actually propagates, right? We have the resolver and then we have the root server. The resolver knows the root server. They exchange their cryptographic keys. Now, what happens is let's say if they need to resolve a host name, goes to the root server. I'm guessing that the root server in its reply is going to give the address of let's say the TLD server.

DAN YORK: Yup.

UNIDENTIFIED MALE: The public key of the TLD server, both of it would be encrypted by the private key of the root server.

DAN YORK: Correct. That's the process. Basically what happens is…

UNIDENTIFIED MALE: Then I get to know the public key of the TLD and then – okay.

DAN YORK: The stars that were given on here is if I sign BigBank.com, I generate another record. It's called a DS record. I upload that to the registry – the .com registry – who confirms it's valid, and then they sign it with their key, and then previously, the .com TLD was signed and they gave their DS record up to the top to the root zone. So there's all this different chain down there.

If you come on Wednesday morning, we have some deployment maps where we'll show all the TLDs that are out there that have signed and then have what's called a DS record in the root zone, and there's a significant number of them. Not all, but a significant number

So for instance, for Fiji – what's your TLD?

UNIDENTIFIED MALE: dot-fj.

DAN YORK: .fj? Okay, I don’t know if .fj has signed. Do you know? I don’t know either?

UNIDENTIFIED MALE: [inaudible]

DAN YORK: I’ll look. I’ll look in a moment. But if .fj has signed, then they will have a DS up in the root, and then you could sign your .fj domain, and then you could give the registry the DS record that they need to make this global chain of trust.

Now, giving that key is one of those little details. It’s often registries are – oh, Roy wants to jump in this – registrars play a role here in many cases.

ROY ARENDS: .fj has not signed.

DAN YORK: Okay.

UNIDENTIFIED FEMALE: How come so many countries have not signed?

DAN YORK: Well…

UNIDENTIFIED FEMALE: Repeat the question, please.

DAN YORK: So okay. The question was, “How come so many ccTLDs have not signed?” Sometimes it’s a question of awareness. Sometimes it’s a question of tools. A number of the ccTLDs use tools that don’t fully support the signing of some of those. In some cases, the TLDs are actually being operated by somebody else and the service provider needs to sign them. It varies, but the challenge with the signing side, you get into a little bit more demands on infrastructure, just in terms of keeping the singing up to date and keeping the operational processes going.

So sometimes it’s not an unwillingness. They haven’t got there quite yet. Roy, as an operator of some TLDs, perhaps you can give some insight?

ROY ARENDS: Well, no, just basically a statement on this. If top-level domains decide not to deploy DNSSEC – and I’m not talking about those that are trying and I’m also not talking about those that will do it in the future – but there are some top-level domains that actively decided not to deploy DNSSEC and said so in public. What they are actually doing is denying DNSSEC for their entire country. That means that end users do not have a choice.

At Nominet, a long time ago, we’ve decided the choice is not ours to deploy DNSSEC. We will deploy it in order for registrants to make their own decision if they want to deploy it or not. If we don't deploy it, no one down the tree will be able to do that.

DAN YORK: One of the interesting dynamics—

ROY ARENDS: Almost done, sorry.

DAN YORK: Oh, sorry.

ROY ARENDS: You said a lot of theories. Actually, of the 628 top-level domains currently in the root zone, 442 are signed. So it's actually a minority – less than 30% that's currently not signed.

DAN YORK: I was going to say to that point one of the dynamics that's happened in this last years with the launch of the new gTLDs, all the new gTLDs have to be signed with DNSSEC by default and they have to support that, and registrars who want to support new gTLDs have to do this piece about passing these DNSSEC records up to the registry.

So with the new gTLDs, we've seen a renewed push in that, so a lot of those that are not signed, a lot of them are some of the country code TLDs that are out there, and I think we'll see an increased pressure on them to sign in some cases, and an interest in signing around that.

I know that a number of organizations have been going on doing training programs in a lot of the regions where they are not signed yet to help people understand the value and to make it happen.

UNIDENTIFIED FEMALE: We have a question here, and then there. Here, there.

DAN YORK: Okay.

[NAVID KHAYAT]: Hi. My name is Navid. Actually, I want to understand, are we talking about making the data secure or making the communication channel secure on which the data actually travels? Because that can make a difference, right? So what DNSSEC is about? Are we actually securing the data, or we are just trusting the person who is sending the data? So, thank you.

DAN YORK: We're securing the integrity of the information. So what you put in the DNS as the operator is the same information you're going to get out as the person asking the question.

We're not securing the communication channel between the resolver and the authoritative server. Dr. Evil could still conceivably get in there and try to inject the packet, but the validating resolver is going to know that it's missing a signature, and it's going to go and do that.

Now, there are other efforts on the way looking at ways to go and secure – to use TLS or to do other ways to go and secure that communication channel. But DNSSEC is about the integrity of things.

Roy wants to jump in here.

ROY ARENDS: DNS is kind of funky. Actually, the data in DNS defines what the infrastructure of DNS looks like, so you can't really look at DNS and DNSSEC and ask if it's the infrastructure that you're securing or the data you're securing because you're actually doing both by signing the data. If you sign the data, which defines the infrastructure, thereby you kind of secure the infrastructure, if that makes any sense. Thank you.

DAN YORK: Yeah, it is a bit – oh, Russ, you want to say something?

RUSS MUNDY: I do.

DAN YORK: Russ wants to jump in here.

RUSS MUNDY: Actually, one of the cool things about DNSSEC is that the transport doesn't matter. If the original person signs the object, it can go person by person, and as long as they recite the right hex codes along the table and then give it to a carrier pigeon, fly it across the city, then none of that's secured, right? Somebody can shoot the carrier pigeon. Who knows what could go wrong? But as long as it gets there, it's still validatable, so it really doesn't matter how many insecure hops it goes through in the middle because in the end you know the object you received was the same one that was originally sent, no matter what the path was.

DAN YORK: Yeah and there's some other pieces we're kind of glossing over a little bit in there to make sure that the validating resolver knows that, "Hey, there should be a signature here. Wait a minute, I got a package without a signature. There's a problem."

UNIDENTIFIED MALE: Thank you, Dan. I'm [inaudible] from Morocco. I'd like just to ask about these public keys that will be used in these infrastructures. So we notice when we use asymmetrical cryptography that the big deal is with the public keys. So how can we trust that the public key is the real public key of these websites or the server?

DAN YORK: Sure. This goes back to the global chain of trust as far as that when I sign it with my public key, I'm going to create this record that I pass up to the TLD, which then gets signed by the TLD, who has then been singed but the upper level. So if you're the validating resolver and you get back this record – this DNS key record – you're going to be able to track that up the root; up this global chain of trust, we call it. So you'll be able to know that is the key. That's how we can be able to go and do that.

RUSS MUNDY: Well the thing that I wanted to point out is just as it is critical when you set up a recursive resolver that you have a root hints file that tells you where to start to look for the root name servers. It is just as critical that you have the public key for the root to be able to do the validation because it comes down for that. Now, there's multiple places in which the root zone public key is published, and these are fairly widely known, and there's a bunch of different ways to get it.

For those that are concerned that they have the right one, they should check two or three of these different places and make certain that they get the same answer when they go to look, "Okay, is this the public key that I should have? Yes." So it's a good idea to check a couple.

DAN YORK: I saw the gentleman in the white back there. Julie, if you want to…

WILL: I'm Will from [Hosting Island]. You mentioned about what's required for the ISPs and it's relatively straightforward by the sounds of it. If you are a DNS provider, such as a hosting company, is it as simple for them to implement, and if it's complex and is requiring to redo keys, etc., is it something that you're expecting hosting companies or DNS providers to sort of charge their customers for? Do you see that being something, or is just going to be a free service just to make the Internet work better.

DAN YORK: The answer is, I think, it depends. There are some DNS hosting providers who charge for this that, if you want their secure DNS services, you pay a little bit extra more for it.

Others just put it in for free. There's some in Brazil who just sign every single domain that's hosted for them, and that's just it. They do it on that.

There is a bit more work on the DNS hosting provider side because you do have to do the signing. The interesting part is, any time the zone file gets changed, you need to resign that part of the zone file. So when a user goes in and they update the DNS address for their web server, you have to resign the zone. If they go and update their mail servers, you got to resign the zone.

So there's some operational aspects that a DNS hosting provider has to work through and understand. It's not difficult. People are doing it all over the places. The good news is that, from two years ago, the tools now have made it much more automated.

We were just in the tech day where they had a panel of I think six or seven different DNS authoritative server implementations – BIND, NSD, [inaudible] all these different ones were in there – and all of them offer real-time or online signing that just happens. So all of them can be set up to do this automatically. But there are some things. You have to set that up. You have to think about key expiration. What happens when a key expires? You don't want to be that Nasa.gov that winds up not being accessible to people, especially now.

I guess the message I would say to hosting providers like yourself is now's the time to do it, to try it out, because the penetration of validating DNS servers is still relatively strong. It's 8-10% overall. In some countries, it's much larger. In some countries, it might be 60 or 70%. But still, it's a good time to try out your systems and get that working before you get to the point where Facebook signs and suddenly you can't get to Facebook or something like that.

Okay. Other questions? Yes, I see the gentleman over here, perhaps. Actually, yeah, feel free to grab that, then I'll get to you.

RJ GLASS: Hey, I'm RJ Glass with America at Large. I just have three really quick questions.

DAN YORK: Okay.

RJ GLASS: First off, how does the user know if DNSSEC is being used?

DAN YORK: The answer is right now they don't. What happens in DNSSEC – and actually, this is somewhat of a challenge – is it gives back what's a serve fail. It just says, "Can't get to the website." Can't get there. So they don't actually know. The 18 million customers in Comcast where I'm from in North America don't have a clue that they've got this added protection. They just know that they can't get to websites sometimes, and they might think it’s done or something, but it might be because they couldn’t get there because it’s not safe.

RJ GLASS: Well, what I’m asking is whether or not the domain resolves or not, there is no way to tell whether or not you’re using regular DNS or DNSSEC.

DAN YORK: No. When they created DNSSEC, there was not a separate error code put in that says this failed validation. A couple of folks are talking about how we could bring that in to try to look at that, but it’s not there.

I will say, for those who are DNSSEC enthusiasts, or people who are interested, there are some things – like I have a little plug in running in my Chrome browser that shows me when I go to a website is DNSSEC enabled. I’ve done and done that. And if you go to one of the lists on the back there, you can find these kind of plugins that can do that.

The Bloodhound browser that I mentioned that the DNSSEC Tools Team has will go and show you that visual indicator. The folks at CZ.NIC Labs have come up with a couple for Firefox and Chrome and IE. There’s some plugins that can show that. But the average person’s not necessarily going to call those. But if you’re interested, you do have that capability.

RJ GLASS: Cool. Also, he was talking about the carrier pigeon routine, right? So yeah, if that carrier pigeon gets shot down, somebody’s got access to the envelope that’s inside, right?

DAN YORK: But all it is is – yeah.

RJ GLASS: If you combine that that HTTPS or something like that –

DAN YORK: Yeah, you can, but again, it’s signed by the private key, so somebody else, if they intercept that packet or anything, it doesn’t matter because it’s signed to the private key that they don’t have access to, so there’s no way they can rewrite that.

RJ GLASS: The whole packet then is signed?

DAN YORK: Well no—

RJ GLASS: Packet by packet?

DAN YORK: The records are signed with a signature record in there, so you have like a couple of A records, and then you’ll have a signature for the A records, and a couple of Quad-A and then a signature for those. So those pieces are signed. There are signatures for those. So the whole packet isn't signed per se, but those records are. This is zone data again.

RJ GLASS: Okay. I gotcha. Last one, real quick.

DAN YORK: Sure.

RJ GLASS: We talked about the browser, basically. E-mail. Let's say two people are e-mailing back and forth. One person may be using DNSSEC through their MX record. The other person may not be. What's the danger there?

DAN YORK: Well, the danger could be that the person who does not have DNSSEC validation enabled might wind up getting a record for the wrong MX record and send the e-mail to the wrong server. So they might get that back. Now—

RJ GLASS: So if I send something to you and mine's secured, yours is not, and then reply back to me, that entire message coming back could be vulnerable.

DAN YORK: Could be vulnerable if somebody where to hijack the DNS records and tell my e-mail server to deposit them somewhere else.

Now, your mention of e-mail brings up another really interesting thing that's happening in the DNSSEC world. There's something called the DANE Protocol. Anybody here of the DANE Protocol? No? Okay. A few people there. You might have heard a few things there, Wes.

The DANE Protocol is a very cool use of DNSSEC because what it does is it lets you put a TLS – an SSL – a TLS certificate into DNS and sign it with DNSSEC, either a full certificate or a fingerprint of a certificate. Okay, it says a hash of that there.

What can happen – and this is why e-mail flags it; people are using this – is you can use DNSSEC as the mechanism to get the keys that I'm going to use to have a secure communication with you using TLS.

RJ GLASS: Both sides?

DAN YORK: Both sides. So if you want to go and send me e-mail, all right, and you and I are both set up with DANE records – it's a new TLSA record, etc. – but what happens is, you will get from DNS the certificate you need to communicate with me. I will get the certificate I need to communicate with you. They'll both be signed and secured by DNSSEC rolling up with global chain of trust so we will now absolutely how we talk to each other and how we encrypt the communication between each other. It's a beautiful thing. It's being rolled out in the SMTP world.

Wes, help me out with the mail servers that are using that. I know there's Postfix, and there's a couple others that have announced recently that they're adding – I think the Exim folks is going to.

WES HARDAKER: Going to.

DAN YORK: There's some mail providers who have been rolling out DANE encryption. There's XMPP, Jabber – the Jabber community. They've all gone and done this for a lot of the public Jabber servers are all now secured by using this to set up server-to-server communication, and in some cases, client-to-server communication. There's people in the voice of IP world who are using this now to provide a way to use DANE and DNSSEC to get secure records for this.

In the web world, there's add-ins for browsers and stuff that will work for this, but we're not yet seeing the implementation in browsers. But the more people see it and use it, I'm hopeful the browser vendors will start to – they worship at the altar of speed, so they have to be very concerned that this will not impact their speed. There's people working on ways to get around that.

Did that help?

RJ GLASS: Awesome, thanks.

DAN YORK: Yeah. You, sir?

UNIDENIFIED MALE: Okay, just a quick one. I'm kind of frowned by the caching mechanism that tends to exist in the DNSSEC process. In relation to the information that is being cached within the database, did you think that it poses like a serious problem for end users, and does the caching process have anything to do with the information of the end users?

DAN YORK: By the caching process, you mean when I was talking about the caching of records as it goes through?

UNIDENTIFIED MALE: Yeah because you said the caches are poisoned.

DAN YORK: Right, right. That's what DNSSEC can guard against because when you get that record – because if Cath is the ISP, she's going to cache the data from the zone, and also the signatures. So she'll have all of that secured information to pass back to Joe User in that case. So she'll have that.

But this caching that happens is part of the challenge of why DNSSEC came about because we realized that in order to have this securely, we have to guard against this type of thing.

There's other mechanisms people us with doing very short lifetimes on the records and other pieces like that, too, that they also add into that effect.

Is that good? All right. Yes, sir?

UNIDENTIFIED MALE: [inaudible] actually. I'm a ccTLD operator and I decided to sign [inaudible] keys are – I'm just configuring my key and publish it and I can get the key automatically the root zone key and [change] it? Are there a manual process, actually? [inaudible]

DAN YORK: So, yeah. You sign your keys for your zone and you have some mechanism where you're signing those and keeping it up to date. Where I say "you," it might be you – going back to what Russ said – it all depends on what you're doing and your level of involvement with DNS. If you operate your own DNS servers and you want to operate your own signing, there's tools out there, like Open DNSSEC, like some other tools, that will go and do all that and automate it for you. So it will take care of all that and work on that.

Now, the one gotcha is getting that – if you change your signing key, if you change that, getting your updated DS record, your signature, up to let's say your TLD if you're a second level, that process right now is still not entirely – there's a couple different ways you could do it.

There's some ways you can do it using EPP (the Extension Provision Protocol). If your registry supports that, you can use that to go and do that. Sometimes it's a manual copy and paste between web forms. So that part is still something that's being worked on, and Warren Kumari, who's not here, but Warren and Oliver and actually Wes have all worked on drafts within the IETF that are looking at automating this process a bit more so it's less manual than it is today.

This is one of the things that when we rolled out DNSSEC we're getting out there. As the deployment happens, people are starting to say, "Oh look, we need to fix this one piece," or do something like that.

Go ahead, Russ.

RUSS MUNDY: Just one addition to this. When the best practices were looked at for how often you ought to change your keys, there is an RFC published about that, and it has time frame recommendations in it.

But on the other hand, there is no established scientific cryptographic information available that says you ever have to change the keys that you're using, because most of the time, cryptographic weaknesses that require new keys are because of how long they're used, how many times they've been used, and what they're doing.

And the what-they're-doing part is normally focused on secrecy, not integrity. Since DNSSEC is straight integrity, there are some people in the community that argue you should never need to change your keys. So that is a position that some people have taken, that key rollover/key change for a sort of regular, ordinary zone is something that is only needed just to keep the people in practice, not for cryptographic purposes. So you can use the key for a long time, if you choose.

DAN YORK: You'll also notice if you look at that best practices document Russ recommended it's actually a common practice people use to have two keys. There's one key that you use that you create this record that goes up to make this chain of trust, and then there's another key – a zone-signing key – that you use to sign your records, and then you sign that second key with the first one to go and do this.

But the net of it is you can change that second key whenever you want to, and so you will often find people might change that one on a monthly basis or a quarterly basis and they'll change the other one maybe once a year or maybe not, as Russ mentioned. So there's some pieces in that best practices document.

UNIDENIFIED MALE: [inaudible]

DAN YORK: No, the keys can be changed automatically by the software. I use a DNS hosting provider that their software just automatically every 30 days – oh, sorry, the question was, "Is it a manual process to roll the keys?" and the answer is – well, I guess it depends on what software you use, but it doesn't have to be. There's software that will do that all automatically, and I use for many of my domains I use a provider and they just e-mail me and tell me, "Hey, your keys are about to change," and it just automatically does it and they just take care of all that.

Now in one of the cases, they're also my registrar. They're the DNS hosting provider and the registrar, and so when they roll my key, they just automatically communicate to the registry. So in that particular case with that provider, my entire involvement from the user in signing my domain was to check a box. That's all I had to do. I checked a box and they took care of it all for me.

Now, in another case, I use a DNS hosting provider who rolls my keys and stuff, but they don't have a way to communicate to the registry. They're separate. They're not a registrar. They're a separate entity. So with that provider, I have to do the little manual jig to go and do that.

Sorry, you have a question?

NEAL SMYTH: Yeah. Neal Smyth from HSBC. So where you've got hundreds of domains dealing with passing those keys up, that's a significant challenge for enabling DNSSEC across that portfolio.

My other question, rather than a statement, was I noticed in the skit that the chain of trust was established between the root and the .com and BigBank, but there was no chain of trust established between the ISP and the user.

DAN YORK: Yeah, the question was there's no chain of trust established between the ISP and the user, and you're right. This gets into a question that we've had within the DNS community, which is, "Where is the safest place to have validation running?" and the safest place would really be in your operating system, honestly, on your laptop or in your device, because then you know when you go and type in www.BigBank.com you know that the validator is running on this laptop, and so there’s no chance that somebody could intercept it.

Now, if you’re in an enterprise and you’re probably running your validation on your internal DNS servers that are on the edge of the network, and then the zone of attack where Dr. Evil could swoop in is really just on the enterprise network, and so you may have to trust that your enterprise network’s not going to be attacked. It’s less of a chance.

Similarly, if you’re talking to your ISP, odds are probably pretty good that your connection between let’s say your home and your ISP is probably pretty secure, if you’re just from your home to there. It’s probably good.

Now, if you’re out a WiFi café or at this hotel or somewhere else, the zone goes out there, which is again where people have a little bit of concern with Google’s public DNS for instance, or the other providers. Yes, you’re getting validated answers, but you’re connecting across the public Internet some great distance to get that information, so Dr. Evil could still swoop in there and be able to provide that.

So the safest answer is put the validation as close to the end user as possible. In our grand scheme of where we’d like to see it, we’re excited that the ISPs have it, and we’re excited the ISPs are doing it. What we want to see next is it being brought down to the home network level – the CPE devices that everybody has, the home routers and stuff. We’d love to see it getting involved in there. We’d love to see it getting down onto the operating system.

By the way, if anybody's a Fedora fan, the next version of Fedora that comes out is going to have DNSSEC validation on by default. So it'll be done there in the operating system already in there.

Russ is waving about mobile devices. Those are also important.

RUSS MUNDY: Mobile devices are important, and there is DNSSEC available for some mobile devices. So you can do it today, right now. Maybe not a full spectrum, but you can do DNSSEC on mobile devices now.

DAN YORK: That's a good question, though. As far as you've got hundreds of domains, yeah, Comcast, when they did their validation that they rolled out, they also wound up signing 5000 some odd domains that they have. I don't actually know how they're dealing with it all. They might operate their own registrar. I don't know what they do, but it is a challenge on that.

I see a gentleman over here, and we've got time for probably one or two more questions after this.

UNIDENTIFIED MALE: It's just short one. Basically, it's a positioning question. As far as the browsing is concerned, my question is how do we position DNSSEC as compared to HTTPS when you have a secure HTTP? I know that somebody cannot spoof me if it is digitally signed and all that, so do we need DNSSEC along with HTTPS, or what add-on it would provide –

DAN YORK: They solve two different problems, okay? Because the HTTPS solves the confidentiality problem to ensure that somebody can't sniff that information and do that.

But DNSSEC solves the step before, which is, "How do I know that I get to the right sever?" because before you connect using HTTPS, you have to do two things. You have to get the IP address of the server you're connecting to, and then you also have to get the certificate from the server that you're going to use to encrypt that connection.

DNSSEC can actually help with both. In its raw form, it helps with that first instance to make sure that you're talking to www.BigBank.com. It tells you that you're getting the correct IP address back. You're able to do it. So that would help you.

Now we'll know the right place. You can connect there. You can get the TLS certificate. You can start up HTTPS and you can have confidentiality now.

So one is integrity, making sure you got the info, and the second one is this.

Now, DNSSEC with DANE gives you an even higher level of trust because what you can do is you've got the IP address to get to, you're connecting to that server, the web server kicks back to you and gives you a TLS certification and it says, "This is the certificate I want you to use to encrypt our connection."

Now, if you're using DNSSEC and DANE, you can check DNS to say, "Is this the right TLS certificate that I'm supposed to be using?" Now you've a super-secure connection because you know that certificate you've got is this because we had people coming to us saying, "Why should I care about DANE? I just spent all this money on this very fancy EV SSL certificate that turns your address bar green. Okay, I spent all these thousands of dollars on this. I'm secure."

So my quick answer is, "How do you know your users are getting that very fancy EV SSL certificate? Because I can hijack your website through DNS. I can set up a site that looks exactly like yours and I can go and buy a TLS certificate. I can social engineer my way to get one that resembles your site. So somebody goes to this site, they see the lock icon – okay, they don't see the big green address bar but they can see lock icon – they think they're going to the right place. But DANE and DNSSEC gives you not only that IP address, but it can also give you the assurance that that is the correct certificate to use.

Gentleman next to you. I think we probably have time for one more, Julie? Is that…

JULIE HEDLUND: [inaudible] five minutes.

DAN YORK: Okay.

UNIDENTIFIED MALE: My question is, would the recent revelations by Snowden that the NSA had a hand in tweaking the cryptographic standards or the random number generators that form as a standard for NIST – so I'm just wondering, because those random number generators are used by a lot to generate cryptography keys, so what's the impact of that on the DNSSEC community?

DAN YORK: Sure.

UNIDENTIFIED MALE: Given that [inaudible] hackers or people with bad intentions, if they know how to predict the random number generators, they might be able to know? We compromise—

DAN YORK: Sure. That's certainly was a concern when all these revelations were coming out within the Internet Engineering Task Force (the IETF), we've spent a lot of time looking at how to strengthen the Internet against this kind of pervasive monitoring. NIST came out with revised guidelines about revised random number generators and pieces to use. Different other software vendors and other have checked and rechecked their code and came out with new ways to go and do this.

So I think part of it is the Internet as an ecosystem has responded to that by going back over and examining the code, looking at the new ways to go and do this.

I'd also say too that DNSSEC provides multiple different algorithms that you can use to sign your keys in different ways so you as the signer can evaluate which one you want to trust. You have a choice in algorithms. You're not told, "Thou shalt use this one." You can sign with different higher levels of securities, different algorithms. There's a variety of choices you can use.

So certainly we feel within the DNSSEC community that there's a pretty high level of assurance that that's not an issue.

I'd also point out, if you're interested in seeing the level of diligence that goes into this, every quarter ICANN performs what's called a key signing ceremony, where they go through a whole process where the gentleman was. They roll over one of the signing keys and they go and do this, and there's a highly, very tightly scripted, very tightly audited process involving locked rooms, sealed things, keys that are sent around by many numbers of different people. So there's a very involved process in ensuring the security and integrity of that process.

I think we're certainly—

UNIDENTIFIED MALE: [inaudible]

DAN YORK: Yeah. If you're really interested, you can go and sit and watch the video of the last ones that have occurred, too. So it is out there, as far as that. You can go to whatever the ICANN root DNSSEC key is up there. Julie's been to a couple ceremonies, I think.

JULIE HEDLUND: Actually, I've participated in all of them since the very first.

DAN YORK: Ooh, I didn't know that. So Julie has watched it. So you can ask Julie—

JULIE HEDLUND: No, no, no. I don't watch it. I actually have role, yeah. I have a key role. I won't tell you what it is, though. But I think you can find out online, actually.

DAN YORK: All right. One last question here.

UNIDENTIIFED MALE: [inaudible] Now, when UK.com signs and gets a new public key generated and it sends it to root zone, if the root zone still has the old zone file and you get a [inaudible] push the new sign zone with that key, is it still going to respond to queries, or is going to—

DAN YORK: Yeah, there's a whole process defined in an RFC 5011 that talks about the rollover process that goes on. So it's typical that what happens is that when a TLD is going to go and have a new key, they'll actually run both keys. They'll do dual signatures for a certain period of time, and then they'll age out the other one. So there is defined process so that the zone would never go insecure, so you won't have that. The TLD will step through a process like that.

Okay. Anything else? All right. You've been here – oh, wait. Wait. One last one. Here we go.

MICHAEL BLUTH?: Michael Bluth, student. I'm still unclear as to how the request would know that it's got to be a signed domain. If it goes downwards from the root, so from the top-level domain, the fully qualified domain name is verified, but how does the root know or the top-level domain know it should be [signed]?

DAN YORK: It's actually the resolver. How does the resolver know any records coming from BigBank – part of it is this DNS record that goes up a level in each one. They'll see one of those floating around from BigBank.com, so if they wind up getting an unsigned response, a good detection will say, "Wait a minute. I was supposed to get a signed result." Because otherwise, yeah, an attacker could just go strip out the DNS and send a bogus thing in there. But there is that check built in through those records.

All right. Thank you all for staying here in this hot room. We appreciate you all. Please do on the resource there. A couple of us – Russ, myself, and Cath – will be around for a bit more, and we welcome your questions.

Again, if you're interested for a more detailed dive, come see us on Wednesday in Hilton 1-6, where we'll have all sorts of different fun and exciting topics there.

Thank you very much.

[END OF TRANSCRIPTION]