CR - DNSSEC for Everybody
Monday, March 12, 2012 – 16:00 to 17:30
ICANN - San Jose, Costa Rica

Julie Hedlund: … to participate and we'll be starting in just a couple minutes. Thanks.

So thank you everyone, please do come in, if there's spots up at the table, and it looks like there still are, please come join us on the table, and I guess we'll – yes, I think we'll start in just a couple of minutes.

]background conversation]

Simon McCalla: Good afternoon everybody, welcome to the DNSSEC For Everybody Session, thank you firstly for coming. It's the purpose of this afternoon is a really brief whistle tour of DNSSEC, and it's designed to give you a chance, if you've never really understood or never really thought about DNSSEC before, or wondered, I don't really understand it, or I'd to learn a bit more, this is the ideal session for that.

It's a light-hearted hopefully, but a fun session. Hopefully, by the time you've left here, you're got a little bit more idea about DNSSEC, what it means to you, what it means to your organization. There is a sheet, it should be on your table here which kind of explains the session, tells you who is presenting, and on the back there is a list of really useful resources, if you're new to DNSSEC, and you want to find out a little bit more after the session.

We won't be stopping and starting through the session, we'll just be going. So let me introduce some of our speakers, right now. My name is Simon McCalla, I'm from Nominet, so we run dot UK. We're very lucky to have with us some of the world's DNSSEC experts in this room, of which I am not one of them.

But we do have Matt Larsen from VeriSign who heads up the research and R&D at VeriSign. Thank you Matt, for joining us.

We have Roy Arends who is also from Nominet, one of the co-authors of the DNSSEC standard. We have Russ Mundy from Coleman and Sparta who are very much part of helping to get the tools and technologies out for deploying DNSSEC.

And we have Norm Ritchie from IC, of course authors of buying to the world's most prominent DNSSEC resolving software. The DNS Resolve It.

So we've got a really great selection of experts here. Do please feel free to ask questions as we go and stop us if there is something that you don't understand.

So let's start, many of you probably think that DNSSEC was invented in the IETF a few years ago, and it was a very technical standard, but actually that's a bit of a lie. And DNSSEC was actually invented 5,000, well actually 7,000 years ago, back in 5,000 B.C.

I want to introduce you to a lady called Ugwina, she's a cavewoman and she lives on the edge of the Grand Canyon. She's very pretty. On the other side of the Grand Canyon is her boyfriend, he's called Og. And he lives in a cave around the other side of the Grand Canyon. The problem these two have is it's an awfully long way around, and it's a heck of a long way down, so they don't really get to talk very much. So one day they get together after they trekked around the edge of the canyon, and they get together and they noticed the smoke that's coming out of Og's fire and they start to cook up an idea.

And before you know it, they're sending smoke signals regularly to each other across the canyon. And they're chatting away quite happily and all is good in their world. Unfortunately, along comes another caveman and this guy is pretty naughty and his name is Komansky. And Komansky moves in next door to Og, and he's got a bit of a thing for Ugwina too. So he starts sending smoke signals. And now the poor girl is really confused. She doesn't know who's trying to chat her up.

So she sets off down the Grand Canyon to go and sort out the mess. When she gets to the other side, she says hey, we need to talk to the village elders, and sitting in the camp is a wise old village elder called Diffy, and caveman Diffy thinks he might have a pretty cunny idea of how to solve their problem. So he dashes off into Og's cave and it was like what's he up to.

And inside of Og's cave is a big pile of sand and the thing about the sand in Og's cave is its funny color, it's blue, and it's the only cave on the edge of Grand Canyon with blue sand in it. So he's got an idea. And he rushes out with a handful of that blue sand, and he throws it into the fire, and sure enough the smoke turns this fantastic blue color.

And now Ugwina and Og can chat away happily. She knows every time she sees blue smoke that's Og chatting away to her, and every time she sees the gray smoke, well that's Komansky and poor old Komansky is confused, and doesn't know what to do, and his evil plot has been foiled.

And really that's all there is to DNSSEC at a high level. So if you remember one thing from today, it's about that blue smoke. It's about putting something into DNS responses that means you know that the response that you've asked for is the one that's coming back to you. And that's really it at a high level. So if you're just going to remember one thing, and we'll talk about that blue smoke again, remember the blue smoke.

So now we're going to hand over. Roy is going to take us through a little bit more in depth into what DNSSEC really is.

Roy Arends: Thank you Simon. So I'm going to give you an introduction to DNSSEC. I'm going to start off with the high level concepts of DNS.

DNS is basically a whole bunch of zones. Whenever you see a domain name and remember there are dots in a domain name, dots will separate the labels from each other. If you for instance take www.bigbank.com, every name in there is a label, and everything is basically structured as a tree.

First off you have the root zone.  The route zone doesn't know anything about bigbank.com.  The route zone only knows about the top level domain names. The top level domain names like dot CR and dot COM and dot UK; they know everything about the second level domain names, like bigbank.com. Bigbank.com doesn't know anything about COM, doesn't know anything about roots and because this is so nicely separated, it's highly, highly scalable.

There is another system involved and it's called the resolver.  A resolver is a little bit of complex piece because this is the machine that actually does all the work.  A resolver will start at the root zone and the root zone will delegate the resolver to the next level and so on and so on.  And this goes on, it's called recursion.  And this goes on until the question is answered.

This goes incredibly fast and there are an enormous amount of DNS servers out there, and an enormous amount of DNS resolvers out there.  And resolvers have been optimized to be even faster by using a cache.  A cache is basically where all the information is stored.  You can imagine if you've been to bigbank.com before, you don't have to go there again, because it's unlikely that in a short period of time this information has changed.

So remember this picture Ugwina, the resolver in this case is chatting with Og the server.  In our case, as you know, Ugwina the resolver will chat with many Ogs, some Ogs are more uglier than others, but in general this is how it works.

So I'm going to introduce you to a small play.  We're going to have a small play to show you how DNS resolving works.  For that I'm going to introduce again Simon McCalla, he's going to be the root zone.  We have Matt Larsen; Matt is going to be the dot COM zone.  We have Russ Mundy; Russ Mundy is going to be bigbank.com.  We have Norm Ritchie; Norm is going to be Joe User.  And in this play, I'm going to be the ISP.

So Joe User, our real world user in this case, can we go a little bit to the left for the screen, thank you.  Joe User in this case, he wants something.  Joe?

Norm Ritchie: Hi, I'm going to do my banking, so I'm a home user, and I want to go to www.bigbank.com so I can do my banking. I'll hand this off to my ISP.

Roy Arends: Thank you, Joe. I have no idea where www.bigbank.com is. My caches are empty, so I'm going to start at the root. Root, I want the address record for www.bigbank.com.

Simon McCalla: Hi Mr. ISP, well I'm afraid I can't tell you where bigbank.com is but I can tell you where dot COM and dot COM is 1.1.1.1.

Roy Arends: COM, I need the address for www.bigbank.com.

Matt Larsen: Well, I don't know the address for www.bigbank.com, but I can tell you that bigbank.com name servers are at 2.2.2.2.

Roy Arends: Hello, bigbank.com as 2.2.2.2, I want the address for www.bigbank.com.

Russ Mundy: Well, I have that address. That address is 2.2.2.3 for www.bigbank.com.

Roy Arends: Thank you, now I have this information and I'm going to cache it for future use, and I'm going to give that information back to Joe User.

Norm Ritchie: Thank you Mr. ISP. Now, I will go off and do my banking at 2.2.2.3.

Roy Arends: So this in essence is how DNS works. It has been working like that for almost 30 years. Next year the DNS protocol will be 30 years old, and what you just saw really happens in real life. And it's after you type in the domain name in your browser for instance, and before you see anything, so this is incredibly fast.

This is how it works. But it is so old, when it was designed, there was no security in place, and these names can be easily spoofed and caches can be easily poisoned. Now, that's a problem, and the reason for that is, DNS uses UDP and not TCP. TCP is an internet protocol and that uses a form of a handshake. It's like when you call someone up, when you call someone up, you state your name, the other side give their name, and basically you have a handshake. And every information you send back and forth gets an acknowledgement.

Now, with UDP, it's somewhat different. It's like writing a postcard. You write a postcard, you put up a name there and an address and you post it, and you just turn away and you never look back. Hopefully, as some point in time, someone will send you a postcard back.

The real problem is you can't actually authenticate just by looking at an address who that person is. So we go back to the caveman story from Simon, Ugwina, the resolver can actually be easily confused, because she can't tell who the real Og is. We're going to show you that again in a little play, and if I could ask the players to, so this will look an awful lot familiar as before, but there is a slight catch in it.

Norm Ritchie: Okay, Joe User, I have more bills to pay. I'm going to do some more banking. So Mr. ISP, I'd like to go to www.bigbank.com.

Roy Arends: Thank you. I've cleared my cache, so I have no information I only know where the root is. And the root is of course at 0.0.0.0. Hello root, I want the address for www.bigbank.com.

Simon McCalla: Hey Mr. ISP, I’m afraid I don’t know that address but I do know the address for dot COM and it’s 1.1.1.1.

Roy Arends: Thank you, dot COM, I want the address for www.bigbank.com.

Matt Larsen: Well, I don’t know that specific address, but I can tell you that bigbank.com’s named serves are at 2.2.2.2.

Roy Arends: Thank you. So I know where the bigbank.com name servers are and I will go to bigbank.com to ask for www.bigbank.com.

Thank you bigbank.com and I now have the address for www.bigbank.com. And I have no way to tell if it really came from the real bigbank.com.

Norm Ritchie: Thank you Mr. ISP, now I can go off and do my banking at 6.6.6.6.

Roy Arends: So again, this is how easy spoofing is. It really that easy, it goes just as fast as DNS itself and a solution for this problem is called the DNSSEC. Now it gets a little bit more complex now, so bear with me, I’ll try to do my best.

DNSSEC uses digital signatures to assure that the information is correct and came from the right place. Cryptography works something like this. You have two keys and these keys are related to each other. One is a public key, that’s the key you give to everyone, and one is a private key. And the private key you don’t give to everyone, you keep that to yourself very, very closely. Because everyone knows that you give them that key, everyone knows that for instance bigbank.com is associated with that key. I or bigbank.com can use the private key to sign something, and the whole world can now use public key to validate it.

Cryptography is basically a bunch of keys and a bunch of signatures. And the public keys and the signatures are just pieces of data, just like an address record or a text record or NS records. And the cool thing about DNS, you can now store these keys and these signatures in DNS, and you can look up the data, just like you look up address records and text records and stuff.

There is another step that needs to be put in. In order for the resolver to trust the root zone, there needs to be a transaction first. So we call that implementing a trust anchor, or co-figuring a trust anchor. Now, the root's trust anchor is very well known. If you want to see a validation in your resolver you need to basically configure that and by default then it will do the DNSSEC validation.

But also between COM and root, and between bigbank.com and dot COM, there needs to be an authentication step as well. So at one point in time, the keys that bigbank.com uses for their zone needs to be authenticated by dot COM. And when that is done, the dot COM zone will place a DS record in their zone, and a DS record is a simplified version of the key that bigbank is actually using.

So in order to show that, we're going to do a play again, and this time – I'm sorry, let's first go back to the slides. Remember Komansky, Ugwina, the resolver, can now verify that the real Og sends the message. That the real Og, the real name server has to send this message and DNSSEC can be used to validate that. So could I ask you to come to the stage again?

So first up is all of these zones needs to authenticate each other to their parents basically. So COM needs to authenticate themselves to root. What just happened here is the DS records being implemented in the root zone, so now we can trust COM. We already implicitly trust the root zone, now we can trust COM.

The same is happening for bigbank.com and COM. And since I now have the key configured for the root dot COM, I can now start validating.

Norm Ritchie: More bills to pay, so I'm going to go off, Mr. ISP, I would like to go to www.bigbank.com.

Roy Arends: My caches are empty, so I'll start again at the root. Hello, root, I need the address for www.bigbank.com.

Simon McCalla: Well, I'm afraid I still don't know the address for www.bigbank.com, but I do know where dot COM's name servers are, and they are at 1.1.1.1. And I'm going to give the certificate.

Roy Arends: Thank you, and by this handshake I have now validated that the information I received is correct. So I can now go to the real dot COM name servers. Dot COM, I would like to have the address for www.bigbank.com.

Matt Larsen: You can probably guess that I don't know it, but I can tell you that the bigbank.com name servers are at 2.2.2.2, here you go.

Roy Arends: Perfect, yet again I can validate that information and it is correct. And now I will go to bigbank.com, bigbank.com I would like to have the address for www.bigbank.com.

Whoa, whoa, I'm getting the address but it just doesn't feel right. I tried to validate it and I can't because the key is wrong. So let me just try this again. I would like to have the address for www.bigbank.com.

Russ Mundy: And the address is 2.2.2.3.

Roy Arends: And so now I can declare victory. We now have the address for www.bigbank.com and Cruella Deville has been defeated. There is the address for bigbank.com.

Norm Ritchie: Thank you Mr. ISP, now I can do my banking at 2.2.2.3 and thank you very much for deploying DNSSEC.

Roy Arends: So much for the play. Thank you all. I'm going to hand over to Russ Mundy.

Russ Mundy: I guess I'll still be bigbank and 2.2.2.2. Well, thanks everybody, we have a fun time doing that, and hopefully it helps bring home in more human terms something that is very much down in the bits that most people aren't aware of, aren't really very conscious about, but my portion here is to talk about how one can go about implementing DNSSEC, regardless of where you might be in this big large DNS array of things.

And as you probably know, there is many parts that make up the DNS, there is activities that own names, many of us in this room own names, so we're holders of names, or owners or renters of name space in the DNS. There is a whole bunch of activities that are involved in getting those names both into the DNS system and managing them once they're in there, getting them out to people, and so DNSSEC touches in some way all of these things, and so if what you're doing, if for instance your ccTLD activity, changes are very good you have a very high reliance on the viability of what you're doing for your function on DNS itself, so you will probably have either in-house or through a very strong contract relationship, very knowledgeable people that do DNS and know DNS well.

On the other hand, if DNS is kind of periphery to your business, you need it for your website and for email, but it's not really core to what you do, you may in fact have all of your DNS activities farmed out to someone. And so I just like to describe what the set of things that are involved, might be.

Like say ccTLD operator or another kind of operator, big, important, probably knowledgeable DNS staff, major companies such as HP, or IBM have staffs that are very knowledgeable in DNS. If you are a company where it's not that critical, chances are you're just going to use whoever you either registered your name with, many registrars offer as an additional service, running name servers for you, and then there is all us end users here.

Now, here is an illustration of the DNS tree, and you can see that it is very much structured like a tree. Here is the large enterprise of HP and here is another enterprise of CNN. Now, CNN is a big active website, and I'll use it in a couple of more examples, but their core business is news. So they may or may not have a large DNS staff, I honestly don't know.

But if you do have this large knowledgeable DNS staff and do most of the functions yourself for DNS, you'll probably want to do the DNSSEC activities with that same staff structure. If you are farming it out in one way or another, to another activity, you'll probably want to do it like that also for DNSSEC pieces.

So where are all of these pieces in some kind of pictorial form? This is something that I drew a few years ago for an earlier ICANN meeting, trying to illustrate that the content of DNS is really the important part. When we just went through our little play, it was getting the proper IP address to Joe User and not having it spoofed in between. And you saw some of the steps that we went through there, but you can call them the owners, the registrants, or clear over as you see it on the far left-hand side there. You've got registrars, they may or may not be present in every zone, but they're extremely common in I think by zone count many more have registrars than don't.

Registries, every TLD has a registry. You must have a registry to make it work. Then there is the name server operation, and each zone has to have a name server operation. Many times it is tied to an existing service. But people tend to automatically think in some cases that their registrar must be their name server operator. It doesn't have to be the case. It is in many cases, but it doesn't have to be. You have to have someone to operate the name server machinery.

And then on the other side, the DNS resolvers and the actual end user application. So you can see many players are involved. And that's kind of the direction that the content goes, up to the peak and then down to the users on the far right.

So here's another illustration of it. Again, in this case a www site, like www.bigbank.com. Someone has to put the information into the authoritative name server for that, then when the client asks a question over here, Joe User wants to get it out, then a bunch of requests go back and forth and eventually Joe User gets an answer. If you're dealing with a larger enterprise, this is an illustration from – actually Matt Larsen originally did this for an earlier version of this of CNN.com and shows you the size of the constellations of name servers involved. So your packets can go to any one of those.

And this is an illustration using a tool that some of my folks have developed to actually map queries and responses for DNS, and this web page, we did this about four years ago, and that's how many queries it takes to fill that one web page. And so it's not a single query or two or three. If you go to a large web site like that, there are very, very many queries, and the complexity continues to grow. This is about six months ago to the same website. So a huge number of DNS queries. And remember what Roy was saying earlier, that these happen essentially in almost a blink of an eye. How long does it take you when you go www.CNN.com to get the web page filled? Pretty quick, and all of these queries can be going on behind you at that time before the web page actually – or as the web page is filling.

So the important thing that I want to make and continue to make here in this part of the presentation, it's actually the data, it's the data that's in the DNS, the names, the IP address information, that's what really matters. That's what the DNS is all about.

And when you do DNS security to the various and sundry pieces, whatever pieces you might have to do with, an important thing to remember is that it's still the DNS data that's the most important. DNSSEC was created to protect that data. So as you saw in the play, somebody can't just sneak in and substitute data on you, but again it is the DNS data that DNSSEC is protecting. And sometimes there is an overly large emphasis placed on protecting the DNSSEC cryptographic material even to the detriment of not paying enough attention to protecting the DNS data itself.

So in the picture that I showed earlier of all of the pieces, the left-hand side of the picture is sometimes called the provisioning side, it's the side where the data itself for a zone gets put into DNS. And the right-hand side, this is the running on the wire part of DNS. And that's the side of things that DNSSEC protects in its active sense. But even on the left-hand side, there are things that need to be done with DNSSEC.

As you saw or if you remember the earlier handshakes, the handshakes of placing the trust up the tree that has to occur before the running mechanism can make use of getting the data out of the tree. The data flows the same way as it has before and so DNSSEC is on the running machinery side of DNS. So in general as I had said before, when you find your spot in the picture of the whole DNS structure of things, and think about what you're doing, I'll give you a couple examples in a minute here, then you'll probably want to use the same approach you're using today. In other words, if you've got a staff that's very knowledgeable, you probably want to use that staff, if you've got a contractor that does it for you, you'll probably want to use them.

So if you're operating a large DNS operation today, you probably will want to make use of the existing products that you're currently using and incorporate the use of DNSSEC into them. For instance the buying product from ISC fully supports DNSSEC. In fact early last week I saw, I had a meeting with Microsoft and Microsoft is making good progress, they've got a beta of Windows 8 and they look like they've really advanced what they're doing in that product line.

There is many other both proprietary and open source products available out there and there is another exciting activity, at least exciting for me as a DNSSEC geek, and that is signing services, where if what you're doing today in your DNS operation, if for some reason it's just not viable to incorporate a change into your – whether it's your registry or your name server structure for your enterprise, there are what's now being called a signing service, where you can take the data for your authoritative zone, ship it off to VeriSign or Nominet or others and they will do the signing of the data for you, and send the data back to you. So you'll get a larger file back than you sent out, but you won't have to actually make any change to your machinery other than be able to take the flow of the data that was going to your name servers, send it off to the signing service, before it gets to the name server, and what comes back goes into the name server.

And like it says in the slide what you do in your particular instance could be a mixture of all of the above. So if you're using vendor products and I won't mention any more specific vendor products, there's a ton of them out there, we do have a survey of DNS aware products, but there's still many products that aren't DNSSEC aware, and so if you have one of those products and you go to the vendor and say gee, I want to do DNSSEC, and they say golly, I'm sorry, I can't, then you have a choice to make. Use different products, push on the vendor, wait for them, it's not unlike other features that you may have gone to vendors asking for and they say oh, we don't do that yet, but my strongest urging there is to go ask for it, because for at least a number of years, one of the things that we heard in the DNSSEC room was that customers aren't asking for it. So please go ask your vendor for it, and like I say many do, but there are still a fair number of them out there that do not.

Now, it's also possible that given such an environment, again the signing service possibly might be helpful in that instance, but it's a lot less likely to fit easily in than when you're running and in control of your entire operation.

So if you're the owner, holder of batch of names, I have a bunch of names registered, most of them are work-related, some of them are personal. And I have to say I don't even know what my total number of names are and so I'm sure there's other people that have that. I've also heard some corporations have hundreds of names. So you need to again examine what you're doing with your management of your names that you have registered and think about how you get those names into the DNS as signed zones by the authoritative servers.

So if you're using a registrar and they're providing your name service, then you ask that registrar when they can state offering the service of signed name server operation, since you're buying that additional service from them. If you're operating your own name servers, that then becomes like the in-house problem we talked about earlier.

So again it's a similar situation but you need to consider what you need to do in the overall realm of things to effectively get to the point where when the signed data is in the zone it has DNSSEC in it, so that the validating recursive resolver, our friend Roy can get the information he needs to validate back to his end user. So that's the end of the presentation of some of the high level approaches of what you need to do, and I think Simon is going to take over with a question set here, but any of this that you want to direct him to are fine, but I'll pass it back to Simon.

Simon McCalla: Thank you Russ. Firstly, a quick question to the room really before we start our questions. Has what you've seen been useful? Has anybody found that useful today? Have you learned something new? Is there something we should have covered that we haven't covered? Go for it, yes.

Julie Hedlund: Please, if you could step up to the table where there is mics, and anybody who wants to speak, please do use the mics because we are – yes, that's fine, we are recording the session and it will help everybody to hear better. Push the green button.

Pedro: Hello, thank you. My name is Pedro. I'm a local entrepreneur. I would like to know a little more about the DNSSEC. Is it defined in the IETF reviews? Is there any draft related to it? Is it a protocol? Is it currently on the draft level or is already standardized? Thank you.

Simon McCalla: Roy, do you want to take that?

Roy Arends: My name is Roy Arends. Matt and I, Matt who should be next to me at the table, have worked a few years ago on standardizing DNSSEC , basically this version of DNSSEC. So DNSSEC is a standard, it's an internet standard. According to the IETF, they call it a proposed standard. Many of the IETF standards are currently a proposed standard. That basically means that it's fairly well deployed, there's implements, there are documents written about best current practice. If you want to look it up the numbers are 4033, 34 and 35. And feel free to come and talk to Matt and I afterwards, thank you.

Pedro: Thank you.

Simon McCalla: Thank you, yes, question right in the back here.

Male: Actually, the authentication between the top level domain server and underlying server it's done as we've seen, but actually my question is who is (inaudible) the root server itself. As the request is coming from the user to the root server and there is not any authentication on this level, but after that, the authentication from root server to the top level domain server.

Male 2: So in DNSSEC there is no concept of server authentication, but only the concept of data authentication.

Male: As it's still incomplete, yes.

Male 2: Go ahead, sorry.

Male: Right now that request is coming in from the user side and sending to the root server, okay, how to assure that this request is going to the root server itself containing the detail that come of thought output going to on the layer.

Roy Arends: There is absolutely no guarantee that when you sent a question that it will end up at the right server. There is no guarantee. There is no assurance that when you sent a question it ends up at the right server. Thus, eventually you will get an answer back which contains this cryptographic information, which you can then authenticate and so in theory it then does not matter where the information come from, because you can check if it's correct or not. So there is authentication of data, it's independent of where it comes from. In general you will get it from the root zone, or from the COM zone and so on and so on. But in the end you will be absolutely sure you have the correct information or not. So independent of where it comes from, if that makes any sense.

Male: Okay, thank you.

Simon McCalla: How many of you here are either involved in deploying DNSSEC, have thought about deploying DNSSEC for your organization? Is there anybody here in that situation? Great, so we've got a few people. And how far along the line are you in deploying DNSSEC? Are you just thinking about it or have you already deployed? Tell us.

Male: Well, I mean we started; we have outsourced most of our DNS, our domains that we have registered. We ran into some providers that do that, have DNSSEC. So we're migrating part of our domains to other registers to implement it so we can have that outsourced.

We also run some of our own name servers and those we are half way implementing DNSSEC.

Simon McCalla: And how are you finding doing that? Are you finding that straight forward? Are you finding you have good resources available?

Male: Yes, I mean we found information and we've been winging it, but it's been working fine, so that's been working. We're probably push it on our customers, because we provide managed security services, so we outsource the DNS. So we're providing them also part of the service as well.

Simon McCalla: Oh, that's great. How about some of the rest of you? How far are you along that line of DNSSEC deployment?

Male: We're just – I'm literally right back from [El Salvador]. We're just thinking about it. And I would like to ask, somebody mentioned that this possibility to send – to be signed by other and then get it back. I would like to ask regarding that possibility, how long does it take, I mean in minutes, and then how secure is it by itself?

Simon McCalla: Matt would you answer that question? Talk about VeriSign signing.

Matt Larsen: I mean it varies depending on the service, but many of them are very fast, like a matter of minutes, if not seconds. And usually they use a protocol called [T-sager] transaction signature to authenticate the transfer. So there is a secret shared between you and the signing service so they know they're getting your zone, and then you know that you're getting the properly signed zone coming back.

Simon McCalla: And it's just probably worth pointing out that there are a number of signing services popping up around the world now, a number of organizations. Some people are doing it for free, some people are doing it on a chargeable basis, and there are some small organizations and some huge ones like VeriSign, Nominet and various others that are also offering a signing service, so it's becoming a really important part of helping to get DNSSEC deployed and out, particularly if you don't fully have the resources to implement DNSSEC yourself if you're in an organization and the signing service is really excellent step. In some respects a lot of people are seeing it as a stepping stone, it helps them to get on the DNSSEC chain, and helps them to get going with DNSSEC whilst they think about they implement it themselves. So it's a useful stepping stone service as well as something you can use full time.

Anyone else thinking about DNSSEC. Is there anyone here that is thinking about it and is still trying to decide whether they should or shouldn't implement DNSSEC?

Louis [Pulan]: Hi, I'm Louis [Pulan] from [dot QT]. And we're re-engineering all of our software because for about ten years we've had the same software which was mostly manual and not automated. And yes we thinking about including DNSSEC, but right now it's just at the thought process.

Simon McCalla: So are you thinking of doing DNSSEC yourself, are you thinking of using a signing service, where is your thinking taking you at the moment?

Louis [Pulan]: We've been considering both ways. I have some free labor since we're at the University. But I'm not sure if we can do it with them, and so we're looking at other possibilities also.

Simon McCalla: Excellent, excellent.

Male: Yes, one suggestion as you're looking at a re-engineering process. Look carefully at the products, whether they're open source or commercial or whatever to ensure that they do support the current RFC based DNSSEC, because even with a signing service and you're going to run the name servers, the name servers must be able to properly support the current RFCs. So that's one thing that though a signing service is a huge stepping stone to get into it, whoever is operating the name servers still have to be RFC compliant for the name server functionality.

Louis [Pulan]: Thank you, that's an excellent suggestion.

Simon McCalla: Does anyone here have any concerns about DNSSEC or any worries? Does anybody think that it's not a good idea? I mean there many folks here who believe in DNSSEC very passionately, but there are some that may think that DNSSEC isn't such a great idea. So I wondered, if there is anybody here who feels that way or is worried about deploying DNSSEC? Okay, I'll take that as a no.

So let me ask I quick question to our panelists. If there was perhaps one or two tips that you would say to somebody starting out or thinking about deploying DNSSEC, or what would be your top tips that you would give?

Roy Arends: Well, I've been involved in many stages of deploying DNSSEC so I've had all the scars, if you currently want to deploy DNSSEC, don't overthink it, there's a lot of software that can manage DNSSEC, for instance there is open DNSSEC, there are the tools that Russ Mundy mentioned and basically, of course you need to do some engineering on your side, but there's a whole lot of info in the past about key sizes, about frequent signing and infrequent signing or key rollover, basically do what the top levels do, do what the root does, and you're in a safe place. So use the latest [buying] software for instance or the latest NSD software so you can [serve a side zone]. Use software like open DNSSEC for sign-ins and to manage the keys and then you're in a fairly safe place.

Simon McCalla: How about you?

Matt Larsen: And I think I would add that there are many resources to help you. We have on the – maybe I'm stealing a little of your thunder Simon on the back of the handout sheet here, we have just a few. I mean there is really a remarkable about of information written about DNSSEC, about how it works and about deployment, and there are many people who would be happy to help you. There are several of the resources; one of them is the DNSSEC deployment initiative that Russ is involved with. Another is DNS OARC, DNS Operations, Analysis and Research Center, that I'm on the board of. So there are many, many places that you can ask DNS questions or DNS specific questions.

Simon McCalla: Thanks Matt. Russ, how about you?

Russ Mundy: Well I guess I would follow on to both Roy and Matt's suggestion of don't reinvent the wheel. Look for information that's out there, that's available, that applies to your situation. A number of people that we've had interactions with over time seem to spend a lot of energy up front trying to create things that already existed. So please use the resources and there are a lot of folks out there that willing and happily answer questions and make your steps in an incremental sort of way, and don't try to completely throw out your DNS system, just because you're doing DNSSEC. If you're going to replace your DNS system, go ahead and replace it, but don't do a massive change just because you're going to do DNSSEC. If it needs done anyway, then do it, but if it doesn't need done, just incrementally add in DNSSEC to what you already have.

Simon McCalla: Thanks Russ. Norm do you want to tell us a bit about DNS within …

Norm Ritchie: Well, no I was just really going to reiterate what others are saying. It's not an area where you want to innovate. There is a lot of people that have been through this with a lot of intelligence. Everything is thought out; you don't have to invent anything new. And there is a lot of people that are more than happy to help you.

Simon McCalla: So go ahead, tell us about Bind.

Norm Ritchie: Well, Bind is the standards reference for DNS; it's fully DNSSEC compliant, according to the RFCs. In the latest version Bind 9.9 has in line signing in it.

Simon McCalla: Great.

Roy Arends: Just one thing to add, there is from the early days, there is a lot of information and a lot of fear, uncertainty and doubt in response sizes and stuff and it doesn't really add security, it actually will add security and there is no problem with the DNSSEC. The root assigns, COM assigns, NET assigns, ORG assigns, UK assigns, there is absolutely no reason not to do it and there is all the reasons to do it. Thank you.

Simon McCalla: Great, thank you Roy. Does anybody else have any further questions you'd like to cover, anything else you feel we haven't yet answered for you? Please go for it.

Louis [Pulan]: Yes, thank you, I had not introduced myself before. My name is Louis and I'm from Costa Rica. How does DNSSEC handle signing key verification; like when the key has been misappropriated, taken over or expired or something, how basically has been handled within the DNS system?

Roy Arends: DNSSEC is different from TLS or certificate systems. So in DNSSEC there is no real revocation mechanism other than time. For instance a record is signed, has a signature, the signature has a timestamp, it's valid from X to Y. And the idea is to keep it timestamp fairly short, not so short that you have to re-engineer everything within the minute, but short enough that that signature is not replaced. But in general caches will drop information after the TTL has expired. So that's mostly a day or two days, thank you.

Simon McCalla: I think your point Louis, in the event of a compromise or a problem with the key; you would then roll to a new key within your implementation. And that would then be propagated out across the DNS. And so it's not going to be pulling it back, if that makes any sense.

Great. If there aren't any further questions, there is a whole day of information about DNSSEC which takes place on Wednesday. It starts, Julie was it 8:30 I think?

Julie Hedlund: Yes, it's on Wednesday, it starts at 8:30 it's in La Pas C, so upstairs from here and we're going to have regional updates and stories from people who are deploying DNSSEC, and some real world examples. Basically lots and lots of useful information. So please do join us and there is also a lunch at the end, just as inducement.

Simon McCalla: So thank you all for coming. Hopefully, we'll see some of you on Wednesday. We'll be there all day, so there are plenty of opportunities to talk and learn some more about DNSSEC. I just want to say thanks to my co-conspirators for helping with this session today. Thank you everybody. Please do feel free to come and grab any of us after the session. We'll be hanging around for a bit. If there's questions you'd like to ask and we'll be here. So please just come and grab us. But thank you ever so much for coming and have a great rest of the day.

[Applause]

[End of Transcript]