LONDON – DNSSEC Workshop
Wednesday, June 25, 2014 – 08:30 to 14:45
ICANN – London, England

JULIE HEDLUND: Good morning, everyone. This is the DNSSEC Workshop. I'm Julie Hedlund from ICANN staff, and we will be starting in just a few minutes. Please go ahead and take a seat, and if you are a panelist for this morning's first panel, you are welcome to take a seat at the front here. Otherwise, there are plenty of other seats. Welcome, and we'll start soon.

DAN YORK: Good morning. We are going to be getting started with the DNSSEC Workshop shortly. For those who are behind me, you're welcome to come up and join the panel. Only the first little area right here is reserved for the panelists who are coming in, but there are some more seats down along here. You're welcome to come up and sit at the main table. If you intend to ask questions, it's a great place to be because there's mics by everything that's there. So we'll get going in just a moment here.

On that note, we will just get going. Welcome to the DNSSEC Workshop.

Good morning. Welcome to the DNSSEC Workshop at ICANN 50 in London. My name is Dan York. I am serving as the emcee I guess for this morning and for welcoming you all here.

If you take a look at the agenda that you have, we have quite a busy day full of activities, going from now at 8:30 all the way up through 14:45 this afternoon.

If you look at the sheet that you have, you'll see that we have sessions that are intended for people who are new, intermediate, and also a few expert ones. We've got a range of people who are here.

You'll also notice on the back, the backside of this is your luncheon ticket. If you're planning to stay here, we will have lunch for you. It will be served in this room, so we will be sitting at these tables, etc., that are part of this. So please take a look at that.

Let's begin. I will also mention that we will have a number of remote participants, so when you do ask a question, we would ask you to come to a microphone. If you're sitting back in the area here, we do have a handheld mic that will be able to pass out there.

I'd like to thank the program committee that puts together this workshop. We've been doing this for a good number of years now at these ICANN sessions, and the level of diligence that's put into these programs is pretty strong as far as the people that are here. If any of the members of the program committee are in the room – I see Russ. Anyone else? Okay. Russ. Okay, I know a number are here, but you can see the list up on the slides, and if you're in the Adobe Connect room, you will also see it there as well. I see – well, Andre's here, too. No, no. Wrong Andre. Sorry.

UNIDENTIFIED MALE: Cath is over to your right.

DAN YORK: All right. Cath is here, yes. Cath, who we also have to thank Cath Goulding from Nominet for a wonderful DNSSEC implementer's gathering. When we do these events, we typically have a DNSSEC beginner session on Monday, followed by an implementer's gathering for people who are interested, and Cath and Nominet hosted that on Monday night at a nearby pub and it was quite an enjoyable time. So thank you very much, Cath, for that.

Our sponsors of the luncheon today, for all of you who are here for this, are what's listed here: Afilias, CIRA, Dyn, Microsoft, .SE, and SIDN. I would encourage you if you see anybody from those organizations to thank them because it is through their generous donations that we are able to have lunch and keep us all in here so we continue to interact and talk and work and do that kind of things here, so thank you very much to these sponsors.

I mentioned the implementer's gathering. It was quite good. The program, as you see here, we're going through. We're beginning with a panel on DNSSEC activities in the European region. If you are a panelist and you have not yet come to the front of this area, please do so. There's a couple of seats over here marked with a hastily-written reserved sign, and that is where you are to sit. We will start with that at 9:00.

Jim Galvin is going to be coming in to give us an overview of DNSSEC key rollovers and some of the work that's happening in there. It's not Nick Sullivan, it is somebody else who is coming in Nick Sullivan's place, and Julie is giving me the name of John Graham-Cumming from Cloudflare,

who will be here. I'm personally interested in this. CDNs are one are that we need to understand a bit more about how they're going to look at the DNSSEC provisions.

We've got the panel discussion around HSMs, which should be interesting for those folks who are curious about how we go do that. Then Haya Shulman is going to be here to talk about Cipher-Suite negotiation, the thing she's working on there. Then we have a lunch break, and then we'll have an afternoon session which will be a number of different DANE and DNSSEC applications. In fact, we're going to have a live demo. We're going to try that out here and see how that works in the Adobe Connect room as well. As [inaudible] looks at me, yes, we've tried this before. We're going to try it again. We'll see how this goes.

Oh, it will not be in the Adobe Connect room, Julie tells me. Folks in the Adobe Connect room will get the slides. The rest of us in the room will get the actual demo. So we'll see it.

But we've got a couple of good sessions in there I think that you'll find quite fascinating. Then we'll wrap it up at the end with a bit on how you can help.

Just to begin with, what we usually do here is we talk a little bit about what we're seeing in terms of DNSSEC deployment and how it's changed since the last time we talked about this three months ago.

One of the charts I like to start out with is one that comes from a site Rick Lamb's been maintaining from ICANN that shows the number of signed TLDs that are in the root zone. You see this really huge spike. It's

more than a hockey stick. It's this direct thing up there, and of course, we know what this is all about, right? Anybody? What is it?

ROY ARENDS: It's the new gTLD requirements to deploy DNSSEC.

DAN YORK: Thank you, Roy. I was checking to see if people were awake. You passed. That's good. Yes, this is when all the new gTLDs came in. All of a sudden, the rest of the chart, the bumps went away because this is just this big huge thing that we've seen there. And it is definitely having an impact on the statistics that we're looking at all around.

When we look at these next charts, this is the work that originally began at Steve Crocker's Shinkuro organization. We've now taken it over at the Internet Society. Steve transferred that to us, as far as the DNSSEC deployment maps. We grouped these implementation status in one of five states. The experimental one is we know people are experimenting with it. We know the TLD is doing something like that. The Announce is obviously they've put out some kind of announcement.

Then we have a couple of states where we know that zone is being worked with. The zone is perhaps signed, but it's not actually being used. There's no DS in root, and there's a couple of ccTLDs that are in that state. Then in the upper two, the DS is actually in the root, and then operational we know domains are being signed under that.

So if we look at the current count from what we have right now in the database, we have a total of 437 signed domains out of 632. You'll see

that. That was just from this past week when I measured that. One night I'll mention about this is I've discovered that there's a slight detail in the database. It's not tracking all the IDNs correctly, so the IDN count is not accurate. There's more about 94 IDNs in the database right now. But the overall point is that we're really around the two-thirds point in terms of overall domains that are out there right now.

Here's a larger of view of what the world looks like at the TLD level, as far as what's been signed and not. We'll jump through a couple of those. Here's a larger map. And these slides are available that you can go and see them if you'd like to see. This one lists out the individual domains that are there and who is in what state.

In Africa, we've had a couple of newer ones coming online, so we're starting to get a few more spots filled in on the map. You'll note that if you were at the Africa strategy session yesterday, there was a good discussion there around the work that's happening to go and sign and get more DNSSEC activity happening within that region. So there's a concentrated effort happening there.

In North America, we're all pretty much signed. I'm not quite sure how Greenland wound up in North America as I look at this, but hey, I guess we'll take it.

The Latin American and South American region, we've got some good effort happening in a number of the different ranges, and this is what it looks like today.

The AP (Asia-Pacific) region, we're there as far as movement in what's happening in a lot of the parts in this space.

In Europe, we're very strongly seeing a lot of DNSSEC happening at the ccTLD level. So it's all moving along pretty well in that regard.

What we're kind of looking to do with this next realm of these maps is we're trying to take this and go a little bit further because now we've got these pretty pictures that show the TLD level what's happening, but we're trying to get a better understanding of what's happening at the second level. So people who have been on the DNSSEC coordination mailing list and calls will know that we've talked about this as a way of trying to look and understand a bit more. But this is where we'd like to go with this.

I'm also working on trying to figure out how we can visualize the status of the genetic TLDs because these maps are great to show the ccTLDs, but we'd like to be able to show a similar kind of thing for that.

We're also working on making the code available for people who are interested in seeing what's there, using it improving it – anything like that.

You can get these maps. We publish them every Monday morning. They go out in an e-mail early in the morning, and you can subscribe to them at InternetSociety.org/Deploy360/DNSSEC/Maps. You're welcome to join. It's a public mailing list. Every Monday morning you get a message that goes out there and it has the latest set of maps. It also has the CSV files that contain the DNSSEC status of all the generic TLDs and all the new gTLDs. So part of my fun has been that I'm getting to enter in all the new gTLDs and look at all these names and see where we're going with this new experiment in expanding the domain name space.

One other note just before we go into some other pieces. We mentioned a couple times ago about the DNSSEC history project. This is a project that's been going on for a couple of years. It started originally out of some requests from Steve Crocker and has gone on. It's a project to kind of document the history of what happened with DNSSEC, how we've gotten to where we are.

We are open to contributions. It's a wiki. Anybody who would like to contribute, there's instructions on there. We'd be glad to have assistance in helping make this happen.

That's all we're going to have on setting the stage for what's going on. Any questions about this kind of material or what we're doing during the scope of today?

I will also comment for the people in the room that if you have not seen these shirts, Ann-Marie from .se, for people who are local, they can see the shirt that we have here. You can go to DNSSEC-Name-And-Shame. And yes, that's a real URL. You know, actually – is it signed, Anne-Marie?

ANNE-MARIE EKLUND-LOWINDER: I really hope so.

DAN YORK: I really hope so. Quick, someone check.

ANNE-MARIE EKLUND-LOWINDER: It's not mine. I'm just –

DAN YORK: I know, I know. It was a project that came out of the hack-a-thon that was done with – who was it all? It was with the TheNextWeb and Verisign and NLnet Labs and – somebody else help me out. Who else was it? Was it those two? Well, it was a hack-a-thon that was at TheNextWeb. A number of cool projects came out of them, including one that's in the demo sessions later today. So there's some interesting pieces out of there. Somebody took this very nice picture of Anne-Marie, too, and she has now provided the T-shirts. Oh, Anne-Marie?

ANNE-MARIE EKLUND-LOWINDER: I can promise you that it will improve even better because when you are signed and testing your domain, you will get a happy face. It's not there yet.

DAN YORK: Okay. So we're going to switch now and talk a little bit about some of the challenges and opportunities in DNSSEC deployment and usage. Two years ago, I was here giving presentation talk about sort of what we'd been seeing from the Deploy360 Programme within the Internet Society around what were some of the issues that were preventing DNSSEC from being more widely deployed. What were some of the pieces that were there?

So I thought it would be useful to kind of revisit that two years later and talk a bit about where we've come from, some of which is good news, and some of which we haven't really changed. But let me take a look at that.

When we were talking about this two years ago, we said, “What should the end user experience be?” Two years later, we still don’t really know because in some cases the general opinion is that the end user experience of DNSSEC should not be. There should be no experience. It should simply be they cannot get to the websites.

Other folks would like to see some kind of lock icon or some other kind of visual indication that’s there. We all know that the plugins are available for various different web browsers, etc., to be able to do this and you can make that experience there. We’re still trying to figure out what this is.

There’s also some work underway, or some folks are interested in doing some work to perhaps provide another DNS error code instead of just providing back a serve fail, providing some mechanism that we can know and be able to provide this. But we’re still working around what this end user experience really should be.

The good news on the DNSSEC validating resolver side: two years ago there was not that many people providing validation, but now we’ve seen large deployments across much of Europe. In South America, we’ve seen them. In North America, we’ve seen them. We’ve seen Google Public DNS being a big help. I see Geoff Huston here, who’s been doing his measurements and showing –and he showed the dramatic impact that that had one Google’s public DNS started to do that. So still more work to do obviously – a lot of work – in getting more ISPs doing the validation, but we’re moving along in this space.

The application developer side. This was something we pointed out that needed some work two years ago. Today there’s a good number of

libraries that now have DNSSEC capabilities in it, as well as the new GetDNS API that we'll hear about a little bit later today as well, but it's providing again a mechanism for doing that. So we're seeing some good news in that regard.

On the registrars, this is one of the big challenges we saw a couple years ago was that we had a lot of people who wanted to go and sign their domains but they couldn't go to their registrar and do it.

We have to admit the 2013 RAA has been an assistance here because it's mandated that registrars provide some mechanism for providing DS records up to the TLDs – or DNS key, depending upon the TLD. So that has definitely increased. A good number of registrars who did not previously provide any means for doing DNSSEC information now are in fact providing that. So we're seeing some good movement here. There's still a number of registrars who seem to be somewhat clueless on that, but yes, Roy?

ROY ARENDS: Thank you for that. It's a brilliant list. What I'm looking for is the inverse. So they can look up the top-level domain and then find the registrar. Is that possible?

DAN YORK: You want to look up a top-level domain and find a registrar that will do DNSSEC?

ROY ARENDS: Yeah. For instance, now on left-hand side, you have registrars who do DNSSEC, and then the top-level domains that they provide DS records for, if that makes any sense.

DAN YORK: Yes.

ROY ARENDS: I would love to see the inverse as well as a simple extension of this list.

DAN YORK: Sure. This is list is actually maintained by ICANN, that they've been the ones doing this list. But I agree, it would be a good thing to do to provide that. Do we have a way of easily getting the registrars associated with a TLD?

ROY ARENDS: Well, the information is all there, so it's just a matter of reversing the [inaudible] value list.

DAN YORK: Oh, inverting the form.

ROY ARENDS: Yeah.

DAN YORK: I'll talk to – is Rick Lamb in the room yet? Rick is in the back of the room and he has happily noted that, so stand by for v2 of this page, coming to you by the next ICANN, maybe, Rick?

RICK LAMB: Sure.

DAN YORK: Sure!

RICK LAMB: I don't see anything stopping [inaudible].

DAN YORK: All right. Rick Lamb from ICANN is the one who has maintained the list, so he says he'll be able to do it. Hopefully eventually this list will go away and we won't need to have it anymore because all registrars will do that.

ROY ARENDS: Well maybe then we have another project for Anne-Marie Eklund-Lowinder for name and shame the registrars who do not offer DNSSEC.

DAN YORK: We actually talked about that. Rick and I were actually talking about the idea that we could take the list of all the registrars off of ICANN's page, and the ones that have said they signed the 2013 RAA. Then you could a little red or green column right next to it as far as whether they are

actually providing DNSSEC or not, or something like that – oh, Rick wants to say something. Rick Lamb?

RICK LAMB: One of the things unfortunately I've noticed is that just because they support you, there's a wide variation on how they support. Some it's, as we all know in this room, some is just e-mail, and some it's e-mail many, many, many, many times until they finally do it. Some have a web interface and all of that. I'd be interested in some simple metric there, as well. Anyway, thanks.

DAN YORK: Yeah, exactly. I went to one registrar that supported it, but the web interface was pretty [inaudible]. I'm like, you know – anyway, so we'll see.

By the way, please, as Roy did, feel free to raise questions in here. This is meant to be sort of a discussion around where we're going with this.

JULIE HEDLUND: Dan?

DAN YORK: Yes?

JULIE HEDLUND: Oops. For the record, we do provide that kind of list for .se registrars, so you can tick the box, and if you like DNSSEC, you will have the registrars who are supporting it.

DAN YORK: Oh, and here. Moving along to the user experience that registrar and DNS hosting to that very point, we mentioned two years ago that a few of the registrars had done something where you could just check a box off. Others had you add more forms. Others made it a lot more challenging.

There's still work to do in this space I think in the user experience in general. There's still a lot of copy-and-paste that goes on. We'll talk a bit more about that in a moment as far as the experience of going up to the top level.

One of the challenges we've certainly seen is the websites have gotten much more complex as they continue to evolve. Russ has year statistics in the DNSSEC for Beginners session. I'm looking at Russ Mundy. You said that in the year or two that you done your survey, it jumped from – yeah, one of the sites it jumped from 70 to 150 queries or something like that?

RUSS MUNDY: Right. The site that we looked at was a large commercial site. About five years ago, there were 70 queries to fill a page. Now it's about 120. Same top URL, but there's that many different URLs name lookups that you go through on the page to actually get the complete image in front of the human being on the browser.

DAN YORK: Right. So we had just identified this is certainly one of the added challenges. Oh, Wes wants to say something, too.

WES HARDAKER: One thing that Russ didn't mention yesterday – and this graph shows it, too, so I was the one who actually made the program that generates the graph. The green lines you see are actually DNSSEC-secured resolutions. There's blue ones, which are not; and green ones, which are. Unfortunately, blue far outweighs the green. But the fact that there are some green – and this is years ago – was actually a good sign.

DAN YORK: That is indeed good. The point of this was really that it's not enough just to say that you've signed that top-level domain, whatever the people go to, what the user sees. It's all the underlying pieces as well that need to be signed to go and truly make it a secure website.

Two years ago, there was less awareness of DNSSEC information. I think collectively we as a community have done a lot better in the last two years in getting information out. A lot of the people around this table that I see have published documents, have put up websites, have added content, have done things that have gotten better. Obviously it's part of what I do with the Internet Society, but it's part of what a lot of the other people around here do as well. So I think we're getting a lot better with knowledge out there.

There's still a good lack of information in the broader world. When I go out and talk to people in the general IT space, they're less familiar with it. One area I think that we collectively need to work a bit better at is helping provide more materials for the enterprises, the CMOs (the Chief Marketing Officers), the marketing communications kind of people, the enterprises, the business makers to try to understand a bit more that is there. I think a number of us have found DANE to be an effective way to move that discussion forward, and we just need to continue with some of that work that's going on.

Speaking of that, two years ago there was a real question around, "Why DNSSEC?" Yes, there's the Kaminsky Bug. Yes, there's issues like that. But there's kind of in a large scheme of things less of an issue.

What I think has happened here is DANE has really helped us a lot with that because we've been able to change the discussion from, "Why should I care about DNSSEC? I've already got this EV SSL certificate," to being able to say, "Well, how do you know people are actually using that EV SLL certificate that you paid for? How can you wind up adding an extra layer of trust to things?

So DANE has certainly helped a lot here, and we've seen some large-scale deployments with DANE in the jabber community. The XMPP community has now secured much of their public infrastructure using TLS, and as part of that, a significant number of the servers have used DANE to provide an extra layer for server-to-server or client-server communication.

We're seeing some interesting pickup in the SMTP world through a lot of the work that happened with Postfix and with Exim and some other

different mail sites. I've been impressed that I've seen a couple mail providers who have actually come out with news releases and blog posts about how they are now providing e-mail secured by DANE and DNSSEC. So to the fact that those mail providers are actually doing that is a pretty cool instance of where we're seeing what goes on there.

Obviously, the Heartbleed vulnerability this year increased a lot interest in securing TLS, and so I think we're seeing some good interest finally in that there is a reason for doing some of this.

Since the time of two years ago, I think one of the biggest challenges we've seen has been looking at this question of, "How do we communicate to the parent zone that a new key has been published?" This whole issue of, "How do I get around the copy-and-paste?" If my registrar is also my DNS hosting provider, it's very simple because the registrar can just transfer the new key from the DNS hosting side over to the registrar side, upload it to the TLD, and we're off and running.

It's different if you're hosting your own zone and signing it, or if you're using a third-party hosting provider. Then there's much more of a challenge, and right now a lot of that movement has been copy-and-paste.

If you're not aware of the work that's been happening within the IETF within the DNS Operations Working Group, there's been a couple of different proposals out there. One of them has been primarily work that – well, between Wes Hardaker sitting down at the end, and Warren Kumari and Oliver Goodmanson and [Suresh]?

But the three of you guys have really been focused around providing some different mechanisms right now that are looking at how do we do this. Some of them involved the publishing of a new record that would let parent zones pick that up and see that. But up on the slide, there's two URLs that you can go and take a look at to see, and these are two complementary different proposals that are out there. But this is a big area that I think we're looking at how do we automate this because this is one of those areas that is clearly in need of automation.

The other part has been, once a domain's signed, how do you transfer it between registrars? The folks at SIDN have been doing some good work with extension to the EPP Protocol that will allow you to relay a key from one registrar to another, and that's happening within a new working group in the IETF called EPPEXT or EPP Extensions. So that work is going on. So that's been another piece of work that's been identified and has been happening as we go along with this.

Another challenge I think we've certainly seen has been that once we start getting DNSSEC out there and getting it deployed, there's a good number of challenges in the network infrastructure that have blocked DNSSEC activity. And Wes Hardaker – again, sitting down at the end of the row there – has written with a couple other folks a draft around what he called DNSSEC Roadblack Avoidance. But it identifies a good number of challenges that are there with some of the middle boxes, the NAT devices, the firewalls. Actually, Wes, do you want to say anything about it?

WES HARDAKER: I would love to. Thank you. The draft really is composed of two parts. One is how do you detect that you're in an environment which you cannot function in DNSSEC-wise? Can you get resolution for important things like DNS keys? Does NSEC work? All of those we actually define how you can test for it very easily, and there's a couple of test tools – in fact, one of which I also wrote – to actually do this, where you get little red and green lines to show you whether your network is DNSSEC-capable or not.

But the second half is, okay, if you know you're in an environment where some element that you must do doesn't work, how can you fall back? So our goal is to document best current practices in terms of not only how to test for stuff, but what should you do if you're an application writer or a library writer and if you need to do DNS resolution?

At some point, there are some tools, like DNSSEC Trigger. Eventually falls back to I have to accept ordinary DNS because I've been that bad of a broken environment. Hopefully that kind of environment will go away in the long run. There's a few other things you can do before completely failing.

DAN YORK: Wes is looking for input on this. It's a draft that's in the IETF DNSOP Working Group, so I would encourage people to take a look at this document. Send comments back to either the DNSOP Working Group or to Wes and the authors directly. But it is something that certainly has been identified as one of the challenges for DNSSEC to truly be ubiquitous is that we do have some of the issues that middle boxes that

are eating DNSSEC records or changing things or doing other different pieces around that. So read more in that draft.

The other big issue that we identified this year or the past year that is blocking a lot of the DNSSEC signing by some domains is the issue of CDNs (Content Distribution Networks/Content Delivery Networks – whatever term you want to use) because a lot of sites, and I've personally been bitten by this myself. I've signed a couple of my domains and then I found out that at the end of the day they're a C-name out to a content distribution network, and that network is not signed. So I'm delighted actually that we're having somebody from Cloudflare here to talk a bit about what they're doing.

Within our DNSSEC coordination calls we've talked a bit to some of the folks at [ACMA] and some of the other networks who do have the technology available to do some signing and CDNs of their networks. But there are some challenges they're looking at in how they go and do that. So we'll hear more I think on that a little bit later today.

So that's kind of a roundup of where we've been. I'm hopeful that two years from now we can have another session like this where we talk about it, and we'll see even further activity in some of these.

I would ask the question now before we go on to the panel of are there any other things that people have seen here in the last couple of years that are roadblocks or things that we need to be thinking about in terms of challenges to move DNSSEC forward?

Yes? Please identify yourself, too, for the people who are remote.

CRISTIAN HESSELMAN: Hi, Dan. This is Cristian Hesselman. I'm with SIDN, .nl registry. One thing that might be added to these slides is that there is basically a gap between validation and signing because validation is done by ISPs typically, and signing is done by registrars and they usually don't talk to each other.

So for instance, when a domain name does not validate and generates an error in an ISP, the registrar is unaware of that. So we somehow need to connect the dots in the chain, so to speak. Do you know what I mean?

DAN YORK: That's a good one. Go ahead, Roy.

ROY ARENDS: Cristian is right, and a few like-minded people have thought about this. There's a very nice analogy in the e-mail world, and this is called DMARC. For instance, SPF and DKIM allow you to check for instance if a mail sender is correct, etc., etc. If either of these two checks, SPF or DKIM fails – and the analogy is if DNSSEC fails – then DMARC allows you to send a report back to the original owner of that domain name. Hence, DMARC for DNSSEC allows you to have that report sent back to the original owner of that domain name for which validation fails.

It is not equally implementable, if that makes any sense. You cannot just layer the DMARC thing on top of DNS and be done. But it's an idea that a few of us are playing with. So thank you for pointing that out.

UNIDENTIFIED MALE: Yeah, so we have an experimental system at SIN that basically does this. It interfaces with resolvers of different ISPs at the Netherlands, and it receives validation errors. We send it through the registrars to which these domain names belong. But of course, this is a small-scale thing and you might need to think that on a more global scale how to do that.

DAN YORK: That's great. Excellent. No, that's great. Other comments?

Okay, well with that, I want to again welcome you all to this day-long session and I want to have the panel come up. If you are not already up here, would you please come up here? And I will turn my seat over to Cath.

CATH GOULDING: Hi. Thanks, Dan. So this panel discussion is about DNSSEC activities in the European region. We have six presentations from registries across Europe, informing us how their progress and initiatives are working in their country. In particular, I'm really interested in this. In the UK, it's embarrassingly low, so I look forward to seeing these presentations.

Each panelist has ten minutes, so I'm grateful if you could try to keep to that. We'll probably leave all the questions to the end, but we'll see how the time goes. First up, we've got Anne-Marie Eklund Lowinder from .se. Over to you.

ANNE-MARIE EKLUND LOWINDER: Thank you. I'll try to keep a distance from the microphone because I tend to speak very loud when I get excited, which I always get when it comes to DNSSEC, you know.

So there's no secret that .se is a strong promoter of DNS. It has been for a very long time. Next slide, please.

We're using carrots and we're using sticks over the time trying to persuade people to start using it, and we have been successful when it comes to ISPs actually validating signatures since all the ISPs are doing that. So we're lucky in that sense.

We have a number of registrars who are signing the domains, and they're actually doing it opt out. They're signing all the customers' domains, but if anyone complains, they can get unsigned, but there's no reason for that. So that doesn't happen too much.

We have been working with bonuses for registrars for quite some time now. We came to the conclusion that talking is not everything. Even though this is the best thing ever, it doesn't sell itself. I'm sorry to say that.

So we started to give them some money to convince them that they don't have to put in so much of their resources themselves. So the thing is, for this year, it has been 318,000 domains validated correctly because they have to prove that they do it correct. So if you're singing and something is wrong, you don't get a bonus for it.

So everyone received about 30 eurocents – and that's three Swedish krona – for each signed domain. The largest registrar received almost

half a million Swedish krona, like 50,000 euros. So it's really some money for them to start doing this.

We have 17 registrars who've received bonuses. An additional 59 registrars have signed domains, but they have problems. They don't do it correctly, and that takes us a lot of time because we're looking it up and we're calling them and we're trying to make them do better if they have broken signings. That task takes a lot of time from us, and we have only actually two registrars who managed to sign all the domains correctly. So, yeah.

We're still working with both sticks and carrots. They get a call from the registry support telling that, "No, no, no. You're not doing this right. Please try again."

This is the map of the Swedish municipalities. The green sign means that they are signed and they are doing it correctly for the municipality. The reason why is we have chosen to show the municipalities is that it is a political decision that all of them should be signed mid-2014, and they should do it right.

Now the municipalities have some sort of internal competition that they want to be green. That is the main task. "I want to be green on that map." So that's something.

This is how it looks with our neighbors. Finland has their TLDs signed, but their municipalities? No, not really. "No, we haven't started yet." Denmark has signed, but they still don't have so much green. So we are quite fortunate in Sweden in that sense that we have a lot of working signed domains.

Within CENTR we made a survey last year, and there were 26 European ccTLDs – among the respondents, that is – who had already had implemented DNSSEC. Ten of them were – oops, sorry – oh, it's not me changing this. That is okay. You can go back. It's right. You can go there have a look for yourself.

There's a small consultant firm who had made this map, so I was thinking about Dan's. There is another tool yet to come. He had to sign this. He's using .se's DNS Check tool as an engine. Well, if you want to be on that map, he needs about 1000 euros to add another country and try to make that happen. So if you want to make contact with that person, I will gladly help you with that. Next slide, please.

The DNSSEC bonus I mentioned. We will also have a ten krona or one euro price reduction for registrars who use our campaign offers, which mainly means they get a lower price for the new registration if they sign at the same time. That gives .se like almost half a million Swedish krona in cost – or lost revenue. It's not a cost. It's money that we don't take from the registrars.

Then technically we also assist one of our largest registrars to overcome some difficulties they have with mass signing. They did actually have to stop their signing because they ran into difficulties. We are trying our best to help them sort this out because if they are signed, that will mean that more than 50% of the .se domains will be signed.

When we come to that number, we will probably not in the very long future decide to take more from people who register domains which are unsigned. So it will become even less expensive to register and have

signed [zones], even for the end customers – the registrars. Next slide, please.

One of the things that we have come across and why it's so hard, it is a huge lack of competence among the technical departments who are running the name servers for their own enterprise or municipalities.

And you have the consultants. They are quite good, some of them, but they're not good in DNSSEC. They have to cover a lot of different areas where they have to help out the municipalities. They work for money, period. They're not working for the best of the customer systems. You have to put up really, really hard and specific requirements for the consultants to make them do the right thing. So I don't know. They are probably quite knowledgeable about what's going on in their own environment, but behind the firewall, they are lost. They have no clue on how it works outside on the deep, dark Internet.

Another thing is people are not really worried about the consequences. What will happen if I am not signed? People don't worry about that at all because they think that nothing will happen. So I was just thinking the other day, "Do we really need another Kaminsky Bug to stir things up a bit?" Anyone? No.

But what we're trying to do is educate. We have training classes. We have published recommendation for DNSSEC deployment at municipal administrations, and similar organizations. That is such a sexy title. But still, you can download it from .se's website, www.IIS.se/docs/Recommendations_4_DNSSEC_Deployment.PDF Are you taking notes, sir? No, I will ask you Julie or Dan to send it.

So yeah, that's about it. I'm looking forward to your questions later on. Thank you.

CATH GOULDING: Great. Thanks, Anne-Marie. I think maybe you could replace the stick on the first slide with a picture of you looking really angry.

Okay, the next panelist is Ondrej Filip from CZNIC. Over to you.

ONDREJ FILIP: Good morning, everybody. My name is Ondrej Filip. I'm from CZNIC, administrative domain .cz, and I would like to refer a little about the situation in the Czech Republic. I will start with a brief history.

We [inaudible]. I think we were like [the fifth] TLD that signed. It was in April 2008, and we started with ENUM because we wanted to train a little bit before we went to serious business. .cz was signed in September, and at the end of the month, we opened the [inaudible] registrations so the end users were able to submit the key material into the registry.

As you probably know, root zone was signed like four years ago. After that, we tried a very funny exercise to make key rollovers, so we changed from NSEC to NSEC3. Again, we started with ENUM zone and after we succeeded with that, in August, we also changed .cz.

So we currently run NSEC3 without opt-out because since the beginning we thought that we'd like to have a very high percentage of signed domains, so the [inaudible] [wouldn't] make sense for us.

We have quite the percentage. That's true. Now I would like to talk a little about what we have done for that. What are the key points? Why's it so high?

Okay, so if you're going to roll out such a technology, you need to talk to [your allies] and the closest were the registrars, so we really tried to make things for the registrars as easy as possible so that they could help us with DNSSEC.

We also tried to talk directly with many major stakeholders, ISPs, and governments, and some major websites – some very high-profile websites – that are visible.

We also do a lot of open source development, and of course one part of it also targets DNSSEC and also DANE technology. As Dan mentioned, it's something that really makes sense for many people why to initially support DNSSEC – because of DANE.

We do a lot of PR and campaigning, many public campaigns, and we organize a lot of technical conferences.

So the incentive for registrars from [inaudible] technical, again since the beginning we knew would like to have a really huge percentage of domains, so we did the technical setup of the registry to support DNSKEY – not DNS records, but DNSKEY. So the registrar can just submit one key material for a lot of domains, and whenever they do any operation, like for example key rollover, they just communicate with our registry, just with a few EPP commands, so it's very simple for them and saves a little time. And we have [three other] registrars that has the same key for 100,000 names. So it's quite handy for them.

We also try to market that summer just out to support DNSSEC, so we started a [so-called] certification program of registrar or majority of them. It's not mandatory, it's just voluntary. But still the majority of them entered the certification program. And to be able to achieve five stars, you need to support DNSSEC because it's really tough to make five stars without it. It's quite a huge part of the certification. So the biggest registrars do support DNSSEC.

Also, financially, we didn't lower the price like the Swedish did, but we support registrars with co-marketing campaign programs. So that means that if they have some campaign that is related to .cz or DNSSEC or technologies we do support, we cover 50% of the expenses. But there are some caps, so it [hits] a certain amount. But if those registrars support DNSSEC, we can increase a little bit the caps. So they are motivated to sign some domains.

The open source tools, I think the flagship and the most visible thing was the DNSSEC add-on to many browsers. Now with the new version, we also support TLSA, so the DANE technology. This add-on is for Firefox, Chome, Explorer, and I think I forgot Safari and Opera, so all the major browser platforms. The URL is DNSSEC-validator.cz.

Because we also try to talk to ISPs to start up validating, we face the problem that a small percentage of domains are bogus. The signatures are broken, and those ISPs that were [validating] were disadvantaged sort of because their customers see some of the domains on the market.

So we set up a system that automatically goes through .cz zones, checks all the domains, and if they see some bogus domains we just delete those signatures and warn those people that something is wrong.

We also developed a DNSSEC HTML widget, which is on all webpage, NIC.CZ, which immediately informs you whether your site is validating and your site supports IPv6 protocol. This is quite popular. Many sites adopted this widget.

Last but not least is the brand-new project called Turris. It's a quite huge project. We are developing a secure CPE. We started really from scratch. We developed hardware, put LINUX on it, and it has a lot of security features, including DNSSEC. So it's an integral part of that. And yeah, everything we do is open source, so if you're interested in some of those projects, just check out our website for that.

A little bit about the campaigns. They started I think in 2010 with a campaign called the Good Domain. It was like an IT Crowd-style guy explaining why it's important to have domain and why it's important to have it signed and stuff like that.

Then we started quite I would say a strange campaign, quite a brave campaign. We created videos of people looking like some local celebrities, and they did some rude thing. It got a lot of attention by mainstream media and [inaudible] and stuff like that. At the end of the day, we said, "Yeah it was just fake, but the same thing can happen on the Internet. So secure your domains." It was quite fun and it has huge courage. It was a little bit controversial because those were very famous people. But we got permission from them at the beginning, of course. But it was quite fun.

Now the current project is Internet How-To. It's two-minute education spots in major Czech TV. It covers all the Internet-related things including DNSSEC, IPv6, and it's broadcasted in primetime, so it's very visible. I think our last research said that it covered like 70% of the Czech population, so quite good.

This is my last slide. The current situation: 38% of all Czech domains are signed. I think it's the best number in ccTLDs. All major registrars support DNSSEC, at least those with 90% of market share. So roughly all of them, or all the important. Many major ISPs validate. I just checked this morning the numbers and I found in Geoff Huston's research that not just two or three but also four operators validate. I was surprised. Many important sites are signing, some newspapers, even some banks and companies like that, so it's a very good thing.

And we are very successful in [the beginning], and now DNSSEC is part of the official Czech e-government strategy. It's called Digital Czech 2.0. They stated that every governmental site must support DNSSEC and also DNSSEC must part of every public bit and stuff like that. So that's quite good. DNSSEC is growing in the country.

That's all. Thank you very much.

CATH GOULDING: Thanks, Ondrej. Sorry? Okay, yeah. Go ahead, Dan.

DAN YORK: Just one quick question. Those video spots and things that you have, are they up on your website or YouTube or anything like that?

ONDREJ FILIP: Yeah. I can send you a URL because it's in Czech unfortunately. We did some translations, at least [inaudible] so I think I will be able to provide it, yeah.

DAN YORK: Okay. Obviously they're in Czech, but even so, it'd be kind of cool to show other people to help push that around and let other people know.

ONDREJ FILIP: Yeah, they are freely available because we tried to advertise website form in many other medias, also – on buses and stuff like that – so we tried for people to go there. Not just in TV, but to see it online, as well.

DAN YORK: Very cool. Thanks.

CATH GOULDING: I think maybe we could even set up a demo, or run it at the lunch break. If we could get that organized, that would be fun.

ONDREJ FILIP: Sure, sure.

CATH GOULDING: Okay, thanks, Ondrej. That's really interesting. So next up we have Peter Janssen from .eu. Thank you.

PETER JANSSEN: Good morning. My name is Peter Janssen from the .eu registry. I feel a lot of what I will be talking about is more of the same as Anne-Marie and Onrej just said. Nevertheless, there might be some things left and right which are a bit different with the others. So let's see.

A bit of history. We started in June 2010. We started accepting DNS key material into the registry system. The .eu zone got signed and the DS record got actually into the root servers in the same year in September 2010. If you look at now, May 2014, we have 3.8 million registered .eu names, of which 267,000 are signed, which is almost 7%.

Statistics are statistics obviously, but if the registrar signs one name, he has the capability in principle to sign them all. So if you look at that, that might be a potential of 949,700 domain names that might get signed. Well, they have to at least signed ten names. It boils down to 700,000. If you have at least 1% of the portfolio signed, that's 293,000. So take away from that what you like.

There is some potential there. Registrars have shown that they can do it. Why they don't do it is yet another question, obviously.

What are these challenges then? Well, I think it is a bit of a repeat of what was already said before me. Low end user awareness – people are not aware of what DNSSEC is, what it's trying to solve, or that it exists at all. End users basically don't know that they want it. They don't understand what it is. They don't even know that it exists, so let alone that they need or they want to need it, if that would be an expression.

Obviously there is an investment to do by registrars to actually get some signing infrastructure in place being hardware, being software, being knowledge – all things around it.

There are some priorities for registrars because they're obviously mostly commercial companies and they have priorities in terms of making a living. The new gTLD program is obviously a one-day, although their DNSSEC is a requirement, so it might actually work in the same direction there. But what we see is that registrars are mostly focusing on selling domain names or selling packages and not necessarily focusing on the DNSSEC or the security aspects there.

We've take a lot of initiatives in the past; DNSSEC workshops in general to its registrars, to its end users, to its hosts, to actually spread the word what is DNNSEC, what is it trying to do, how does it work? We have done some e-learning courses which our registrars can follow to actually go through the motions and set up a name server, sign a zone, and see what it is actually like. Training in the most specific way, in person that is done.

We have something called the DSS (the DNSSEC Signing Service). Or rather, we had. It was something that we thought might help registrars to get their zones signed. Basically what we said is, "We set up your [in-sign] zone. You configure a name server that we can zone transfer the zone to our name servers. We will sign the zone and we will push out the signed zone back to you, and then you can move it onwards to your public slaves." So basically we're taking out the administrative hassle of doing the signing, doing the key management, making sure that the public key was in the .eu zone and so on.

Basically, the one and only reaction we got from registrars was, “Cool. Very great. But can you do it for all our domain names, including all the other top-level extensions?” which obviously we couldn’t really do because we didn’t have like the privileged access to the registration system like we had for the .eu registration system.

So basically, registrars have that great opportunity to use our signing service for .eu, but if they want to do it for the whole portfolio, well, they still had to implement DNSSEC for .nl, .se, .alltheotherextensions, and has yet to do it for all the other extensions, anyway. There was no point in using a DSSEC signing service from .eu because they were implementing DNSSEC anyway on their own. So it was nice, but it didn’t work.

The carrot and the stick. About the carrots: DNSSEC discount. Basically, every month we look at all the domain names if it’s correctly signed. So we go through the complete motions from the root servers all the way down to the signed zone and see if all the signatures are correct and so on. If they are, we’ll give them two euro cents per domain name per month to the registrar, which boils down to 24 euro cents a year if they correctly sign their domain names. Basically, we test each month so they can get the two cents a month if it’s correct, or they don’t get the two cents if it’s not correct.

One of the projects that is still coming up is a bit more of the stick side, where we will actively start – well, harassing is a big word – but the registrar is talking about them: “This zone, you didn’t get your two cents because of…” whatever it is they did wrong at the moment in time, mostly expired signatures and things like that.

A bit of history about the DNSSEC discount program. You see on the far left January 2012 and on the far right you see today, so a bit more than two years I guess. You see where we launched the DNSSEC discount program a bit in the middle of the graph, and I think we can sort of decide and say, yeah, the DNSSEC discount program has had some sort of an effect on the number of signed domain names. So money still speaks in this world, apparently, so it might be cool for others trying to do the same.

Obviously, as the .eu registry, we are second in the geographical market. We have the country code top-level domains in each of the European countries and obviously .eu is sort of covering the same territory there.

So one of the interesting aspects is to see if there is a correlation between the successes of an existing ccTLD and the signed domain names there in that extension, and then it goes on to the .eu extension. What you can see is that in Holland, Netherlands, where as IDN has been taking some initiatives lately – well, not so lately, but has been taking some initiatives in the past. And you see that a lot of .nls have signed, but a lot of .eus held by Dutch people or at least registered by Dutch registrars are signed as well. There are some correspondences there.

The uptake apparently in .uk was not that high, but we still have quite some English registrars that actually sign their domain names, so there is a bit of a strange effect there.

I guess basically that's it for me. Thank you.

CATH GOULDING: Thanks, Peter. Yeah, I guess it's interesting to compare across all the European countries with you.

Next up, we have Vincent Levigneron from AFNIC. Over to you.

VINCENT LEVIGNERON: Can I stay here? Yes? Okay. Okay, no problem.

Good morning, everyone. My name is Vincent Levigneron, and I work for AFNIC. My short presentation will focus on results, and the next step of our DNSSEC promotion plan we started in 2013. Next slide, please.

Let's introduce AFNIC in one slide. AFNIC is a French non-profit association, and it was created in 1998. Mainly at the beginning, AFNIC was created to operate six ccTLDs for France and some of its overseas territories. .fr, which is for France; and as far as the overseas territories, there's .re, .pm, .tf, .wf, and .yt.

But now we are not only involved in the ccTLDs because AFNIC has been chosen as the back-end registry for 17 gTLDs. Some of them – and my slide is not up to date – some of them, .paris, .frogans, .ovh and .bzh are already in the root zone, and others will follow soon, I guess. Next slide, please.

A bit of history. I guess my colleagues did more or less the same thing, so you already know about that. In 2007, .se registry was the first to sign a TLD. In 2009 was the start of root zone signing process, and in six months from that date, the root zone will be fully signed and operational and could be [inaudible] from top-level domains.

At the very beginning of 2010, a dozen TLDs are assigned, and many TLDs claim they will sign soon.

In September 2010, AFNIC, .fr, and .re TLDs are signed and are introduced to the root zone. By the end of the same year, all six AFNIC ccTLDs are signed and in the root.

In 2011, AFNIC registries had the possibility to submit DS records through EPP and web interfaces, but one year later, we added only 50 zones signed with DS published and with only 16 registrars involved. That's why we decided in October 2013 to launch a multi-year DNSSEC promotion strategy plan. It's a five-year plan.

So we are at the very beginning of it. I will just give you information about this year because, as you can imagine, we can use feedback of the first year of this plan to improve it and modify it if necessary. Next slide, please.

What have we done in 2013 to promote DNSSEC in our French community? First of all, we have published a practical guide to DNSSEC deployment, which was written in French and in English.

This implementation and deployment manual provides practical guidance for DNS hosts to configure DNSSEC on their infrastructures.

The main purpose of this guide is to provide in short form, because it's less than 30 pages, key [inaudible] to configure, to sign, to monitor, and to debug a signed zone using [inaudible]. It doesn't claim to be complete because it's a short format.

The DNSSEC is not even addressed, but it's filled with common lines and configuration examples. It's free. Anybody can download it if you want, but I'm sorry, I forgot to put the link in my slides. Sorry.

We offer our colleagues a 10% discount for .fr registrars. It was for new domain names and domain name renewed, and we want signed in the five days after the creation of a new one, and this discount was during the October-December 2013 period. It's no longer applied, but we have an Amazon campaigns that will start soon.

We have also a DNSSEC training program in partnership with a company called HSC, who's involved in DNS hosting in France. HSC is a French company that has very long experience in the field of computer security. Next slide, please.

These are the results obtained during the period when the financial incentive campaign was applied. We raised 200 DS per day introduced in our zone. When the financial incentive campaign ended, it was 150 at the beginning, which presents an increase of 25%.

Also this is not [inaudible] of course. We keep [inaudible] of DS registration, even if there is no more discount, which is really encouraging for our future plans. Next slide, please.

These are the 2013 final results. We had a growth of published DS and .fr since the project started of more than 57%. From my point of view, the most interesting result is the increase of registrars who have at least a domain name signed. It is twice the initial number. We have more or less a0 similar result with the number of AFNIC registrars with at least ten domain names signed.

But of course these results mask larger disparities among registrars involved in DNSSEC. For instance, also the growth in the number of [signed] zones is significant. More than 98% of the signed zones are managed by a singular registrar. Next slide, please.

Let's see what it really represents in terms of volume. At the end of April 2014, .fr zone is more than 2.78 million domain names, and almost 5% of the total are signed, which represents 135,000 domain names. 64 registrars have at least one domain named signed, but some of them have only one or two domain names signed, which is 12% of all AFNIC registrars. Next slide, please.

But why [published] campaign was dedicated .fr only zone? DS registration continues on all the ccTLDs. As you can see, some of them have a larger percentage of signed zones and the .fr zone for .re we have 6.5 domain names signed percentage. For .pm, we have 3.5% domain names signed. For .wf, 3.1, etc. The larger one, if I can say, is.yt with 10% while it's a smaller domain. Next slide, please.

This is our 2014 promotion action plan. Our plan is to provide a new revision of our practical guide to DNSSEC deployment with additional parts focused on DNSSEC monitoring tools and configurations. We received very good feedback on the first version of this document. That's why we to invest some time on it, and to improve it, and to meet our registrar's requirement.

We plan to start a new financial incentive campaign with a larger discount. This time, it will be a 20% discount. Again, it's only for .fr domain names and for .fr registrars. While this is more or less the same conditions, this time your registrars are obliged to sign a minimum

amount of domain names to get the discount. Sorry, I don't remember which [inaudible] but I guess it's Dan or [inaudible] I don't remember. Sorry. Next slide, please.

The two other action plans for this year. The DNSSEC training program in partnership with HSC, which continues with two sessions by the end of the year. And something new: we have a DNSSEC HowTo, which has a first live session that will be organized on the 1st of July in one week. Others will follow depending on the demand. The goal is share operational experiences and it's not only for registrars. Everybody can attend. I know people in this room that will attend this DNSSEC HowTo.

The main goal is to propose a [inaudible] and let's participate, meet people who operate everyday signed zones, and there will be a large part dedicated to practical exercises. Next slide, please.

And this is the end. Thank you.

CATH GOULDING: Thank you, Vincent. That's really interesting. Just one quick question from me, because I'm the moderator and I can. Have you set yourself targets for the five-year strategy, or is it just to increase the numbers?

VINCENT LEVIGNERON: In fact, we learned at the same time we started the promotional plan with of a course a target for the end of the five years, but I can't tell you what is the target because it's changing depending on the feedback we have from our registrars. Of course we would be very happy if we could have half of our zones signed, but it really depends on our registrars

[inaudible] ICANN and decide if they want to sign their domain name. So yes, in five years, we plan to have [1000] signed by then. I'm not sure we'll reach that point.

CATH GOULDING: Right. Go ahead, Dan.

DAN YORK: Your session on July 1, is that a webinar or some type of system like that? The DNSSEC HowTo, the live session you have.

VINCENT LEVIGNERON: Excuse me? Can you repeat the question, please? Sorry.

DAN YORK: Is it using some kind of web presentations system?

VINCENT LEVIGNERON: No, no, no, no. Sorry. It's a face-to-face meeting, so if you'd like to attend, you are welcome. So you have to go to Paris – no, not really Paris, but yeah, Paris.

DAN YORK: Okay. Thank you.

CATH GOULDING: Thank you. Okay, next up we have Alexander Mayrhofer from NIC.AT.

ALEXANDER MAYRHOFER: Thank you. Good morning, everybody. I'll speak about DNSSEC in .at and beyond .at since we are doing other stuff as well besides the administration of the .at TLD. Next slide, please.

So where are we using DNSSEC? Where are we providing DNSSEC services? The first service that we actually brought into production obviously was the .at. ccTLD. We have DNSSEC in production since February 2012. As you remember, we were one of the late people on the bandwagon, so to say.

The second service that we provide to our new gTLD customers is called Registry-in-a-Box. It's a registry and DNS service for new gTLD applicants or registries right now. As everybody probably knows, DNSSEC is mandatory, so we are providing DNSSEC services for that product, as well.

The last service that maybe not all people are aware of is we are also a commercial operator of Anycast service/Anycast network. The product name is RcodeZero. We are offering that to TLDs, as well as to registrars as a simple secondary service. We introduced bump-in-in the-wire singing on that product recently. Next slide, please.

The timeline for .at. you might remember you have seen that in 2012 in the ICANN meeting 44. We did a testbed in February 2011. Then we went for the deliberately unsignable at zone, like the root zone did as well, on December 14. We let that run for like about two months. Then we put the DS in the root, and a couple of weeks later we added the

opportunity for registrars to add DS records via the EPP interface on February 29th, 2012. Next slide, please.

Maybe a little bit different to other people, our marketing department was very eager to get a little bit of PR fallout out of that event, so we were probably the only registry who handed over the DS record to IANA staff face to face. We simply had that opportunity during a CENTR meeting in Salzburg.

We did a press release with our first DNSSEC customer, which is Austria.at. It's actually a tourism company. It's not the government. Our marketing department did a bottle of DNSSECCO, so that we shared with the press.

We got about a couple of articles in newspapers and IT magazines. It wasn't effective in terms that people rushed to add DNSSEC to their domain, but it was effective in way of getting the word out about our company, so that was a very valuable stunt. Next slide, please.

A little about the specifics we are doing on the technical end. We are using OpenDNSSEC as software for signing. HSMs – we didn't go for the [Oracle] HSMs, but we went for the Thales HSMs. We have two independent signing and validation chains, so they are completely separate [inaudible] in different data centers, so in case one of the data centers goes down, we still can sign our zone, obviously.

We also got an additional emergency key in the TLD. That one is in the root zone, but it's currently not used for signing. The idea behind this is it's actually in a bank safe – the key materials – so we can essentially set up an emergency zone signing infrastructure with just that emergency

key and whatever hardware we have at hand in case of a real emergency.

We put a lot of effort into validation on the zone, so we actually strip the DNSSEC stuff from the .at soon after we signed it and compared it to the original zone to make sure that everything is correct and so on and so forth because especially in the beginning, we were really afraid about publication of a broken or incomplete zone, obviously.

We have an emergency zone. That zone is the current zone with the serial number one week ahead, signed, so that in case something goes wrong with the DNSSEC signing, we have sort of a zone in the future that can overwrite the serial of the current presumably broken zone. So that's one of the emergency measures that we have in place.

One [inaudible] that we have from the registry side – our domain name transfer optionally removes the DS records. That is the case if the gaining registrar has not yet indicated he's DNSSEC aware. Next slide, please.

Numbers. As I said, I did the presentation in 2012, so the numbers for 2012 are below. In total we have 432 registrars. We didn't get a lot of new registrars because we changed the rules for registrars slightly. Especially we introduced a minimum monthly fee, so that scared off a lot of very small registrars.

Out of those 432 registrars, just 38 have turned on DNSSEC. They need to do that in their registrar web interface. That's still up from 14 in 2012, but as you can see, it's not even 10% of the registrars. Out of those 38 registrars that actually have DNSSEC switched on, just 22 are

actually using DNSSEC, which means that they have at least one delegation with the DS record in the .at registry. That's up from nine in 2012, but still it's very low figures. Next slide, please.

I would like to say we're probably one of the few registrars who did like serial promotions for DNSSEC. Our registrars don't get a price reduction. We don't push our registrars into doing DNSSEC. We just sort of offer the opportunity for them to actually do it, but we don't push it. We did a couple of presentations in our registrar days, but that was about it.

So out of the about 1.2 million domain names we have in the zone as of last week, there are roughly 1000 DNSSEC delegations. That's up from 57 in 2012, but it's very low numbers, as you can see.

Looking a little bit closer at those numbers, it turns out that 800 of those 987 DNSSEC domains are with a single registrar, so all the other registrars have like very low numbers like ten or 20 DNSSEC signed domains. That one registrar is actually a smaller one. He seems to use DNSSEC as a standard for most of his portfolio. So as I said, 80% market share of DNSSEC is with a single registrar in .at. That also clarifies that transfers are currently not really a problem for us. Next slide, please.

That was about .at. As I mentioned before, we are doing registry back-end operator for new gTLDs for .berlin, [inaudible] .hamburg [inaudible] and so on and so forth -- .brussels [inaudible]. And the signing setup is essentially identical to .at, so we essentially copied the infrastructure. But we have separate hardware for that.

The other difference is that the EPP domain name transfer does never remove the DS record because registrars are expected to be able to handle DNSSEC-enabled transfers anyway.

Figures. We currently have seven TLDs delegated. I think that's still correct. And out of those seven TLDs, we have just two signed delegations across all seven TLDs, and those are test domains of a single registrar in the .berlin. So that's about the take-up of DNSSEC and new gTLDs. It's not impressive there either, even though all of the registrars are required to offer DNSSEC. Next slide, please.

The last thing that's probably something different than what most people here in this room do, as I said we have a commercial Anycast DNS service that's called RcodeZero. We are offering that service in two flavors. One is for TLDs, which is just a secondary DNS service for TLDs. We don't do anything about DNSSEC signing there, other than that we are obviously able to publish a singed zone.

The other product is that are offering up to registrars, and on the side of the registrar DNS, we introduce DNSSEC recently in two flavors. One is that they can simply sign themselves and we publish the DNSSEC records. It's quite a common thing. The other thing is that we also offer bump-in-the-wire signing, where they can simply for each and every domain name that they have individually enable or disable DNSSEC, and we are going to do the key generation, the key management, and also the signing of the zone. They can also transfer out the signed zone if they want to use another name server for those zones as well.

We actually are monitoring the registry interaction of the registrar, monitoring in a way that we are looking at the DNS and only doing a key

rollover on our side once the registrar has actually provisioned the new key with the registry. But it's all DNS-based. We are not interacting with our registry system in any way because we are offering that product for all TLDs. So registrars want to push their whole portfolio onto that network. But we have a couple of tweaks in the key rollover process. We make sure that we don't rollover before the registrars actually provision the new key with the registry.

The registry interaction itself obviously remains with the registrar, but are watching it sort of on the distance and waiting until everything is – that's available since Q1 2014 in sort of a testing environment. We've had it in production since two months now. We have a couple of registrars testing that, but it's not like we have a very high number of domain names there as well. In total I think we have about 600,000 domain names on that network, and I think we have like four or five DNSSEC domains there yet. So each and every registrar that signed up for that service is sort of testing it with that single domain. Yeah.

That's about it, I think. Yeah, that's it. Thanks for your time. I'm open for questions of course if the time allows. Thank you.

CATH GOULDING: Lovely. Thanks, Alexander. That was really interesting. Lastly but not least, we've got Sara Monteiro from .pt. Over to you.

SARA MONTEIRO: Hi. I'm Sara. I'm from DNS.pt, the ccTLD from Portugal. Before I start, I just want to say the ccTLD of .pt is a small ccTLD. We have low numbers

compared to all the other ccTLDs, but we work hard, so I hope you like the information that we have brought to you.

In the last year we have had a lot of changes in the organization because actually we have a new organization right now. It's a multi-stakeholder for those who don't know. If you want to know more about it, you can ask me, but this is not DNSSEC related. But I think it would help somehow with some DNSSEC numbers that I'm going to show you.

We started DNSSEC on .pt in 2010 on the 4th of January. The zone roots were signed and we started submitting DS from our end users as well. But we have started working on DNSSEC some years before that, so I brought some numbers about that.

We started I think some DNSSEC information sessions and technical hands-on in 2010. It was just a few workshops and sessions, but we didn't want to just sign up [inaudible] This is DNNSEC, so we started to have some information to our registrars so this way they could just start concentrating on the idea of DNSSEC.

So I just got these numbers, and since now, we can say positively that 500 people have learned about DNSSEC, and some of those were registrars. They can use that information in their system to put DNSSEC available to our end users.

But I think this work, these constant workshop and sessions that are free for anyone who wants to learn something about DNSSEC are very useful. It's hard work, as I said, but at the end of the day, we feel that at least one person on the workshop will implement DNSSEC, so that's nice.

Comparing the numbers of the domain names with DNSSEC, in 2010 we had 60 and right now here we have almost 10,000. That's nice. [Since our zone], there's 240,000 domain names delegated, so it's a good number to us. That happens because some one or two registrars decided that they wanted to implement DNSSEC.

We didn't have any promotion as well as .at. We don't have that yet. It's in our thoughts. We are working on it, but we don't have that scheduled yet. But even though, they like the idea. They wanted to give it to their clients, and they did it. So we are happy that they wanted to be as active with DNSSEC as we wanted them to be.

Some interesting numbers. Our first registrar appeared in 2012. It's a small registrar, and at this time, it has like 150 domain names signed, and that number is t9he same over the years. But they are consistent, and we were happy to see this first registrar two years after our signing.

But right now, we have what we call the gold one that started in the end of 2013: Clara.net. In Portugal, it's known as Esoterica as well. They have a lot of domain names registered – more than 5000.

So that was the boon that we experienced, and maybe because they are more involved in the ccTLD since they are partners on the multi-stakeholder model and they started to work on it, maybe because they want to offer it to their clients. I think both reasons are valid, so I'm happy with that.

Recently, this year I found [this] small registrar. They have like 100 domains and 60 of them are signed, so that's amazing, just starting like that.

So at the end, if I can give you a bigger number, like I know that almost 30 registrars have at least one domain name signed, but the number of one domain name there are so many that I just brought you the top six.

These are the top six registrars. This is the amount of domain names that they have registered. I can say that the 5th registrar as our main registrar. It has the higher number, as you will see in the next slide. The first registrar has almost 70% of domain names signed.

So as you can see, sometimes numbers can be tricky, but we can see that the people that the registrars that are seeing DNSSEC as a landmark and they wanted to give it to our end users, they do it on a large scale. So I hope that all the promotion that we are giving to them and publicizing on Facebook will tease them to want to have more numbers and to be like the green ones in Sweden on the green map. So I hope they will do that.

Even though we have the same problems of all the other ccTLDs in Europe, we have a lack of awareness. We don't know why. Is it only awareness or is it just avoiding? We hope the first one.

I know that if the end user doesn't know what DNSSEC is, they don't have the need to have it. So we need to create the need for them. So we need to put the awareness that they need, so we are planning to do a lot of campaigns. We want to put motivation on registrars. We don't want to go on the mandatory [side], but sometimes some needs need to be mandatory.

We just want to give some awareness and concern on DNSSEC security, and we are handling to know what the best way is, and sharing this with all of you is the best way.

Thank you. That's my finish line. If you want to see more data, you can see our DNSSEC.pt and our Facebook. Thank you.

CATH GOULDING: Thank you, Sara. I think it's clear from your earlier slides with all your training and awareness sessions that you have been working really hard.

Okay, so that completes all the panelists. Have we got any questions from the floor? Do we have a microphone, by the way, Julie?

UNIDENTIFIED MALE: I've got a quick question for you. Sorry.

CATH GOULDING: Hi.

UNIDENTIFIED MALE: I was just wondering, is there any reason why there's no update from the UK of Nominet? Is it because the numbers are that bad?

CATH GOULDING: Go ahead, Roy, if you want to answer.

ROY ARENDS: The numbers aren't that bad. We have [one in thousands] of domains signed, or one per mil, or slightly over 10,000 domains signed. When Nominet was started, bylaws were created, etc., etc., that one of the things we can't do is give any discount or discriminate on any of our fees. At least this was true when we started deploying DNSSEC. So we have never had the ability to promote DNSSEC via a discount towards registrars.

This is still a discussion inside of Nominet and what we can do to promote things. We've done many things in promoting DNSSEC. Nominet has been part of the unbound resolver effort. Unbound is a resolver that can validate DNSSEC. Nominet was part of OpenDNSSEC in order to help implement tools that can do DNSSEC for you. We have met with the folks for ISC to help progress automated signing in BIND, etc., etc. So there's a whole lot of things Nominet does in the background in order to get DNSSEC deployed.

We have talked to ISPs in the UK in order to get them to validate DNSSEC, and some of them are actually doing that. We can see that in our traffic. But indeed, the numbers are low and we really wish they were higher, but this is what it is. It's one in a thousand currently. Thank you.

CATH GOULDING: Thanks, Roy. Yeah, over there?

OLAF KOLKMAN: Olaf Kolkman, NLNet Labs, still. No, no, just to make sure that's clear [inaudible]. I saw a lot of beautiful statistics of numbers, growth curves,

many domains signed. But obviously, that doesn't say a lot about the use and the importance of the domains signed. I'd rather see the Financial Time domains signed than I see my blog post site. My blog post site is signed and the Financial Time is probably not. I haven't checked.

UNIDENTIFIED MALE: It isn't.

OLAF KOLKMAN: It isn't. So there is a bit of importance about what services are behind the signed entities. I'm not looking only at the web. I think DANE and the security that it provides for communication services under the [root server] is immensely important. I think it's more important to have that type of number because it has more weight in my opinion than just statistics of the amount of DS records sold or given away at the registry.

I wonder if at any of the registries there are cases where you say, "I'm proud. This is real content. These are real services that are now being protected by DNSSEC, and this is not a hosting firm where a gazillion domains are parked that now signed but are irrelevant."

CATH GOULDING: Does anybody want to answer that?

ALEXANDER MAYRHOFER: Alex from .at. As you mentioned, it's mostly – sorry about that – it's mostly the key signing their zone, either at the registrar, someone who

believes in the DNSSEC, but it's very hard to push big enterprises. Critical infrastructures I would say – who's calling?

UNIDENTIFIED MALE: [inaudible] DNSSEC.

ALEXANDER MAYRHOFER: Yeah. If you do that to an Austrian phone, actually the wall plug has a small ringer as well, so that's very interesting.

ROY ARENDS: Let's test to see if we're naughty or not.

UNIDENTIFIED MALE: Does the wall plug also have a small microphone?

ALEXANDER MAYRHOFER: Yeah. As I said, we don't really have any high-profile domains signed. We would wish that banks for example would start DNSSEC signing. But when talking to people from IT enterprises, it's interesting that they don't completely always understand it. They think it's like TLS, where you actually prove your identity, while DNSSEC actually doesn't, and it would be very easy to create a high-quality phishing website with a DNSSEC-signed domain, yeah?

So the sad news is that those high-profile sites that we would all wish to be signed actually are not. It's mostly geeks and small registrars, yeah.

CATH GOULDING: Anyone else want to comment on that particular question? I've got loads of hands.

UNIDENTIFIED MALE: I just want to say we are more successful. We have some banks. We have roughly all the ministries and state offices that sign domains, so we are more successful in that.

It's not easy. Usually those sites you need to talk to those guys to be able to explain to them why this is important. So it's a lot of manual work, actually. There is no automated system as far as I know.

[ROY ARENDS]: Then you hit on something very important. You have to talk to these guys. That means you have to convince them of the worth that signing domains brings to them.

What is the story that you told to convince them? I think that is the type of information that I would like to be shared in this forum. How do you get the high-profile users of services convinced that DNSSEC is actually something that improves their security, that brings better services?

UNIDENTIFIED MALE: Actually, the fact that DNSSEC is in the official [state] strategy helps a lot. So that's easy, but there's also one legal reason. There's a potential of attack – I know it's not very high probably, but it may happen – and if such a bank knows that there was a security tool to prevent this attack and didn't use it, then maybe they can be liable for the damages that were caused by this attack. So that's very good motivation for high-

profile banks and people like that to use any kind of security protection, which DNSSEC is a part of it.

UNIDENTIFIED MALE: Can I just add plus one? With all the training that I do around these weird places, particularly some of the developing places, it's usually a government request and the banks are the first ones to start deploying this thing. But usually at the request of the government.

CATH GOULDING: Do you want to say something, Anne-Marie?

ANNE-MARIE EKLUND-LOWINDER: Yeah. I mentioned we have a political decision in Sweden to have both municipalities and agencies sign, but we have this agency who's [inaudible] contingencies agency and we have actually convinced them to raise some funding for the municipalities to actually make it happen, because for small municipalities [in Sweden], this might be the difference between serving milk to the school students or having DNSSEC. They don't have this budget. They don't really easily make it happen. But with some funding from this agency, they have come a long way, actually.

So I think it's important, and what we are telling them is that they are running very fast into what we're calling e-governance, and they have a lot of services online that people – the citizens – of that municipality are using to send very, very sensitive information. So of course they should be signed.

CATH GOUDLING: Sure. Just in the interest of time – sorry, I mean there was about three or four hands on this, but we are already really late. So I was going to take maybe one more question on something else, but this could be debated on the break or at lunch because I think it is a really good discussion.

Is there any more questions on maybe another topic? Go ahead.

UNIDENTIFIED MALE: Yes, I have actually an observation and it's quick. We heard about in Europe about the production side of DNSSEC, but it's interesting that Europe is certainly the highest percentage of users who use DNSSEC. 13% of users in Europe will actually do validation if the name is signed. But it varies a lot.

We heard from Sweden. 75% of users in Sweden will do DNSSEC validation. We heard from the Czech Republic. 45% of users in the Czech Republic will do DNSSEC validation. EU is not a country. Tough. We heard from France. 2% of users in France will do DNSSEC validation. Austria, a little bit better, 4%. Portugal? 1.5%. Great Britain? 8%.

So what it does mean, although we've heard very similar stories on the production side, on the consumption side, there is a vastly different picture. Whatever is happening in Sweden, Estonia, Slovenia, Denmark, Romania, Czech Republic and Poland is not happening in France, Austria, and even Great Britain to that extent.

I think if you push one side, you've got to push the other side. It's the case of using it as much as producing it. Thanks.

CATH GOULDING: Yeah, really good point there. Okay. So I think we should move onto the next presentation, which is Jim Galvin from Afilias.

UNIDENTIFIED FEMALE: Please join me in thanking our panel, including Cath.

UNIDENTIFIED MALE: Well, while Jim is setting up, I will just note, too, that Olaf, since you're here, I will mention that Olaf mentioned he's wearing his NLNet Lab hat here. For those who are not aware, he's joining the Internet Society this next month, and so he will actually be my ultimate boss in looking at the technology side of the house of things. So you'll probably see him back here again asking some of these same kind of questions about how can we move the needle on this stuff.

JIM GALVIN: Okay, thank you. I'm Jim Galvin. Everybody loves a good controversy and I'm always happy to provide, especially when I'm a little bit wrong.

Already this morning, I had my good friend Antoin Verschuren– and I'm pretty sure I didn't say that right, but I'm not Dutch and I'm never going to say that right, but I think he's out there listening – was telling me that what I'm going to propose here as part of the solution is actually incomplete. He's right about that, but I was focused on a particular issue here, so I had kind of overlooked, and I'll get to the point here that's missing when I get there. But I wanted to give him a shout out right away about all of this.

This is something that's been on my mind for many years, actually, since I first got involved in being the part of Afilias and our signing of all the TLDs that we host. We had examined the issue of signed delegations and in gTLDs what it means for registrars to support DNSSEC.

Over the years, a lot of people have looked at transferring DNSSEC services and did a lot of attention to that, and so from a technical level if you look at just DNS and DNSSEC, I think it's a fairly well-understood problem, but there is a particular piece of this puzzle when you're working in the context of gTLDs and the rules by which you are bound and accredited registrars, and the rules by which you are bound that affect whether or not you can do the transfers of DNS services at the same time that you're doing registration transfers. That's really the particular context in which I want to focus the problem here.

Obviously I highlight the transfer process for DNS services just by talking specifically about four particular steps. There's a lot of details hidden in here, but you only need to see this much and understand it for the purposes of highlighting this problem.

We all very well know that ideally what you want to do is create your new zone, get it deployed at your new DNS service provider. The next fundamental issue that you have is getting your keys to move around.

I focus here on the KSK, but in reality to get a complete solution for your DNSSEC transfer, because of the behavior of validators – and we've had many presentations here in this workshop over the years about validator behavior – you really do need to synchronize both your KSK and ZSKs in both zones. This is the issue that Antoin was beating me up about this morning. I'm focused here on the KSK, but you really need to

include both of them. So you need to be publishing both your KSK and the ZSKs in the old and new zones where they belong.

Then you change your NS records. The important point here is you want to leave your old services up for a while because you want to wait for your TTLs to expire and give all of the existing validators for their caches to unload and be loaded up with new information so that you can move on.

This actually fairly well works. We know that. Plenty of people are doing it, so all of this is a good thing if you're just moving your DNS services.

The real problems come in when you look at what the majority of the situation is in a registrar. In a larger context, we always like to talk about the 80-20 rule. You have 80% of the solution or 80% of the problem, depending on how you want to look at it.

In a registrar context, it's very common for registrars to provide bundled services. So you buy a domain name. You're going to get your DNS services from your registrar. You're going to get your web hosting from the registrar and your e-mail from them. It's a pretty reasonable thing to do and everybody does it.

What's interesting in this process is this is actual behavior that we see today from most registrars. If you're going to transfer your registration service, you go to your new registrar, you request the transfer, that request is through the registry, sent over to the old registrar, and there is a five day grace period during which this transaction can be challenged or rejected.

What typically happens is the losing registrar, if you will, will simply notice that a transfer request has come in. They will simply remove that domain name from the TLD zone file because the figure it's going away. "It's not my problem anymore." They no longer provide DNS services, and then they just wait for the five-day grace period to expire. So they don't acknowledge the transfer and let it complete.

Obviously what happens here is the domain goes dark as the TTLs time out for whatever they happen to be set for, usually a day or two. So then you have a day or two of not having services.

The issues that the new registrar doesn't have the ability to make changes to the NS records until after that five-day grace period expires because they don't actually have the authority. They don't have access to the records in the registry to make those changes until they have actually completed the transfer, which doesn't close until the five-day grace period expires. These are just the rules. Nobody's doing anything bad in that context.

So I very quickly took a look at this is how you could overlay these steps and say, "Okay, what does this look like if I take these two things that I describe? How could I lay them alongside of each other?" You still have that wait for the transfer to complete.

The only change I made here as compared to the list of steps that were listed there is, in order to combine it with registration transfers, if you look down the left column and you talk about the pre-publishing of the KSK, you want to say clearly the old registrar in that case, or the new ZSK also, if you were to do both of them.

But if you go to the next slide, it says, “Where are the issues?” Well, it turns out that there are issues in every step of this process on the DNSSEC side. So I’m going to walk a little bit through here what the issues are.

One of the problems is new registrars in the first step there that’s highlighted on the left – deploying the new signed zone – they may or may not do that right away. Usually what they really want to do is they want to own the zone first. So they’re going to take your money and do the registration, and not all registrars will actually stand up your new DNS services right away for you.

Equally important, they won’t provide an export mechanism for you to get at the new key information that you need to carry over. We already know we have an air gap problem here with the pre-publishing of the KSK and the ZSKs, so I have to be able to get them from the new registrar to the old registrar. That’s an existing technical problem. We already know that that one’s there.

The changing of the NS records, the problem that you have at the old registrar is most commonly, if the registrar is providing services to you and they’re providing the DNS services, they actually don’t let you make additional changes there. They won’t let you put the new key information in because they’re the singing authority and they’re doing that function for you. They don’t want you modifying the zone in that way. So they don’t even give you that functionality if they’re the old registrar, and that’s an interesting detail.

They’ll let you change the NS records, but of course when you do that and you’re giving up the zone, then they don’t want to provide services

anymore. So there's no actual step for you. That last step about discontinuing service at the old service provider, if you're just moving DNS servers, we know that you need to keep both services up because you need overlapping services while you wait for TTLs to expire. The old registrar doesn't even give you the functionality to do that. It's not a place where the business has been.

Again, one of the most important things to take away from this is you're talking about the 20% that for which this is critical or essential are probably already doing it right because they probably don't have bundles services. This affects sort of the commodity portion of the registrar business. So the 80% of the mom-and-pop operations that have their domain names, maybe they don't care about this issue, and that's why no one really has been focused on it. But that's why I raised this in this forum. It's an interesting question as to whether or not we want to solve for that community of users or not.

Moving on, the specific functionality that I would assert is needed in the system, okay, is these three things that need to be able to happen. First, I do need to be able to deploy new DNS services at a new registrar. That's probably an easy thing to solve. It's probably relatively straightforward for registrars to decide to do that and add that functionality.

The hard part comes in the next two steps, though, because you're actually looking for essentially the old registrar or the losing registrar to commit to providing services and adding functionality that this is not their customer anymore. This is someone who's moving away from them. So why should I bother to do these things? Why should I let you

bring new key information for another service provider? Why should I continue to provide DNS services for you?

I think it's an important question. It's an issue. It's a business issue for registrars, and the rules certainly in it do not in any way enforce this kind of stuff needs to happen. So what we have is a technology that, as much as we want to ensure that you have a valid zone and you're going to continue to have your valid zone, there's a large community of people out there for whom there's no way for them to make this happen, not if they're going to continue to use bundled services at registrars.

On this last slide here, I talk about what it should look like. I explicitly talk about the steps that need to happen. Some of this actually does affect business workflow inside a registrar. They really need to be able to do their services differently and their functionality different.

And as we are so fond of reminding ourselves in many ways, registrars often talk about having very thin margins as it is in the business that they do. So if you're going to add more work for them, then it's an even tougher sell to convince them that these are the things that they need to offer in order to ensure a better system overall, a more holistic system.

But the critical thing is the things that I talked about here. As you started your new registrar, you need to be able to get the information our – your KSK, your ZSK information – you've got to be able to get that into the new registrar, and the new registrar, ideally, you would want to them to cooperate. Even though the registration transfer is occurring, you want to make sure that they maintain those DNS services, either

until they're told to turn them off or there are actually ways in which you can algorithmically decide.

The usual value that I've heard is you continue the DNS services for at least twice the longest TTL that exists anywhere in your zone that's currently deployed. Most of that's under the control of the registrar, so they know what that is. That's what you need to do in order to get past flushing caches for validators that are out there and ensuring that the new zone will take over.

I think that that's it. Yes? Yeah. So that's the last slide. Thank you.

CATH GOULDING: Thank you very much, Jim. Now I'll open things up to questions.

JIM GALVIN: I'm especially interested in whether there's any registrars in the room who want to comment.

CATH GOULDING: I saw Wes, and then Rick.

WES HARDAKER: Can you do a little compare and contrast quickly between customers that are running their own name servers? Most of your problem statement relied around when the registrars themselves are running the DNS service, right? There's less problem I believe if you are running your own name service. So one possible solution is to transfer your name

service away to somebody for ten days and then do a double transfer, basically.

JIM GALVIN: Yes, you're absolutely right. The context of this problem is strictly bundled services. The assertion that I'm making is, again, 80% of the market, that's what it is. Now, you're talking about your mom-and-pop operations, so in order to know that you can solve your problem by moving your DNS services first to a third-party service provider, then move your registration, and then bring your DNS services back to a bundled service, that assumes that you're somehow technically savvy enough to figure all of that out.

WES HARDAKER: Right. Or move your service to the new provider. Move the DNS service first, and then move the registration.

JIM GALVIN: Right. But even if you do that, you still have the issues of being able to get the keys and move them. Some registrars do that better than others. Most of them, at least in my experience, don't give you key control. They don't give you the ability to enter key records if they're running your DNS and they're doing the signing.

It's probably a pretty reasonable position to have. After all, it's sort of a security issue. But I think that's something to switch.

WES HARDAKER: So then it also makes it equally difficult to, if you want to start running your own DNS service, basically moving away from your registrar and pointing at yourself is nearly impossible because they don't provide that same mechanism. It's the same problem.

JIM GALVIN: Right. True. It's not a perfect solution, but it is a better solution.

CATH GOULDING: Rick?

RICK LAMB: First, Jim, thank you very much for bringing this up. This is actually the barrier for at least one very large ccTLD that has not signed yet. They see this as the fundamental problem – that it's the hosting, it's the ability for their customers to switch between different registrars.

I just was wondering if you had thought of any kind of solutions to this. We've seen various proposals I think in the IETF and things, but do you have any ideas here about maybe a way to simplify this for the registrars, for the mom-and-pop who are just going to sit there any say, "Sign me," and then move?

JIM GALVIN: Yeah. My ideas are on the last slide there. Those are the services that a registrar has to provide. The automated proposals – the work that's going on in the IETF – those are solutions that can also work.

The problem is they don't fix into the context of the gTLDs because of the rules by which you're bound. It's that five-day grace period that gets you and your ability to do things against it. That's really the fundamental issue.

I don't see the five-day grace period ever going away, so you're really going to have to insert constraints inside that and requirements on registrars, or somehow fix that problem.

JOHN DICKINSON: Hi. This is a problem that I know has been around for a long time. It's certainly been talked about [IETF] release the last five years. As far as I can tell, there isn't really a technical solution to doing this. This is an education and regulation issue. I just wondered if someone could comment on whether or not this is something ICANN can educate and regulate and force registrars to cooperate in the handover of a zone from one registrar to another.

CATH GOULDING: I'm sorry, could you state your name, please?

JOHN DICKINSON: John Dickinson.

CATH GOULDING: Thank you.

JIM GALVIN: Well, I don't like to use phrases like "force registrars to do things." That would be bad for my business. But nonetheless, you can make almost a business case for this. If people want to cooperate to provide DNS services validly and completely for their customers, if you're going to do DNSSEC right and you're a gaining registrar, so maybe as part of knowing that you're going to gain some, you have to provide the services on the losing side. So you balance that in your own mind that you're willing to do that for others. So they just see that as a balancing that could come around and they can do it in that way.

Is it a regulation issue? I suppose. And it's unfortunate that that's what it comes down to. Like I said, I think the problem really does exist because of the five-day grace period and the option to not continue services within that period.

JOHN DICKINSON: What about educating the registrants?

JIM GALVIN: Well, now you're back to the comment that Wes was making in the beginning. Sure, you could because, again, you have an almost complete solution if you move it to a third-part provider – move your DNS first and you do all of that.

Can we educate the community to do that? To be honest with you, I suspect not. But again, getting back to my first suggestion, if people want to cooperate to do the right thing, registrars could cooperate.

As a gaining registrar, perhaps I'm going to provide a lot of education to my new customer and facilitate to them that, "Oh, by the way, before you do this transfer, you should go do this first. Glad to have you. Love to take your domain name for you. Go do all this stuff." And maybe there's a service there to help make all of that happen. So that's another path for doing this, too.

It really is fundamentally a business issue, and whether that's regulation or new rules, there are different ways to handle that.

CATH GOULDING: I have a question here from Andy Linton.

ANDY LINTON: Andy Linton, .nz. We went through this process when we decided to sign the .nz domain name. This was one of the changes we made to our registrar agreement. But a set of steps like this is actually built into the registrar agreement, so maybe it's one thing that people need to think about. That's a different question for the gTLDs, but for ccTLDs, it's certainly something that they may have more scope to look at this.

We've got a set of steps that are not exactly like that, but they're pretty close, and when you do the handover, at some point the old registrar gets told, "Now you can drop your delegation and your records," and so on. So it's maybe worth thinking about.

JIM GALVIN: Yeah. An important distinction to make is ccTLDs versus gTLDs because ccTLDs in their own environment, because they can create their own set

of rules and do things that they want, you can solve this problem. This problem doesn't have to exist in ccTLDs because you can do that.

But I think this problem will become more visible as ccTLDs enter the gTLD market. There are quite a number of them now that have applied or gTLDs, and that stuff is becoming deployed. So even ccTLDs will become part of the forcing function to see change in this space because they'll be able to solve the problem in their ccTLD, but not in the gTLDs, and now they're providing two levels of service, and I suspect that will become an issue.

CATH GOULDING: Additional questions? I'm not seeing any more questions, so I want to thank Jim for a very thoughtful presentation and provocative. So please join me in thanking Jim.

Now, you may have noticed that we had scheduled a break. I think we really do need to get back on time, however. That would ensure that at least we would get lunch on time, which I think most people would like. That doesn't mean that you can't step out for breaks when you need to, but we have about seven minutes before the next presentation, and that is from Cloudflare. Is John Graham-Cumming here?

UNIDENTIFIED MALE: Yes.

CATH GOULDING: All right. Well we'll get John set up, but we've got about seven minutes. If you can quickly pop off and pop back, we will start precisely on time at 10:45. Thanks.

We will be starting on time, so I do want people to be here when John starts up his presentation. So just to let you know, you've got a couple minutes to get back to your seats. Thanks.

Everyone, please take your seats and finish your conversations. We are going to start on the next presentation.

We have John Graham-Cumming from Cloudflare on DNSSEC and DNS proxy. I'll just turn things over to you, John.

JOHN GRAHAM-CUMMING: All right. Good morning. Thank you very much. My name is John Graham-Cumming. I work for Cloudflare, which is a fairly large now provider of DNS and many other services in San Francisco, although I work here in London with the company. I am a programmer and I work partly on our DNS infrastructure.

The reason there's a programmer involved is we actually wrote our own authoritative DNS server in Go. I'm not going to go into all the reasons about why we wrote a new thing. We were using PDNS and we switched to our own thing. Afterwards, I'm happy to talk to anybody about the motivations for doing that.

We currently do not support DNSSEC, and I wanted to talk about some of the things that we provide and some of the challenges with DNSSEC at our scale and under the sort of constraints that we have.

The large problem for us is that DNS is quite difficult at the scale we're operating at when we're a large target. What we're a target of is DDoS attacks. We are continuously DDoSed.

For example, right now there is a vote going on in Hong Kong called. It's called PopVote.hk. It's peaking at 300 gigabytes per second DDoS attack against that site. That's typical for the sort of thing we see when people are upset about something, and DNSSEC will add to our woes if we don't do it right.

We would very much like to support DNSSEC. This is definitely not a presentation about, "We're not going to do it." It's a question of, "How do we do it and what's the right way to solve the problems with it."

To give you an idea, this is the top 10,000 Alexa sites of DNS market share. So we are – I think is meant to say the largest. I hate pie charts. This isn't my slide. But we provide DNS for about the same number of people that AWS does.

There are two ways in which we do that. One of them is we are authoritative for domains using us. So what happens when somebody decided to use our services for acceleration or protection is that they point their name servers to us and we handle the DNS for them.

There are also customers of Cloudflare who are ccTLDs or other DNS providers for whom we proxy DNS. That's done to provide protection for that service because they can't handle DDoS. We've seen over the last couple months a couple of DNS providers get into serious trouble with DDoS, so we will actually front that and take the attack traffic.

So there are two different things we're doing. For most people who use us, we are just authoritative, and so we are the DNS server. For others, we are proxying DNS. So we're caching and proxying DNS queries up to the real world. We're not truly recursive because we have special links to those DNS providers.

We also want to be extremely fast. The reason this is important apart from we like everything to be fast is that the thing that Cloudflare sells apart from protection is speed, so your website will be a lot faster if you use us. We care a lot about that.

We do that for two reasons. That was one of the motivations for writing our own server. The other thing we do is we Anycast DNS. So we have 27 locations now around the world. This is the [solve] DNS. We're about ten milliseconds around the world for typical DNS queries that go against our infrastructure.

So we have the challenge of dealing with attacks, and also at the same time being quick because otherwise the customers who are paying us money to give them fast websites are going to be upset.

And we get attacked all the time. Ridiculous things. The DNS is very popular for attacking us, as well as Layer 7 HTTP and NTP and anything you feel like. Depending on how the angry the attacker gets, they will start out with one set of name servers, and then perhaps they'll work their way through our infrastructure and go off to different parts of it.

But we continuously get DNS attacked, so obviously DNSSEC worries us because we're going to start getting attacks. Within the last year, we saw a lot of reflection attacks using amplification, which are just

volumetric, basically. What's happening recently is a lot of attacks that look like purely legitimate traffic. So there are genuine queries against domain names that are on us trying to cause us trouble, so it's not hard to imagine that once you have DNSSEC, people are going to ask us for things that are signed, and that's going to be expensive.

Yeah, so there's two things. The core business is we are a reverse proxy for HTTP and we set that up by taking over DNS for websites, and then we can dynamically change the IP addresses for sites that use us. The ability to dynamically change the IP addresses is very important to us because when attacks happen, we typically advertise a low TTL, and if an attack happens against the web site, depending on the sophistication of the attacker, they may go after an IP address, and we will actually change the IP address of that site and then null the IP address they're attacking. The lower-sophistication attackers just keep going, even though they've been nulled. Others will follow. It depends on how sophisticated they are.

Again, we want agility around the ability to change things, which of course means that, if we're signing things, we have to sign the new things.

We built this resolver in Go, which is called RRDNS. There are two things going on. There is an external DNS server where the world is using to actually get A records and other requests from us. Then there's an internal Cloudflare proxying happening of the DNS records.

So unsurprisingly, something like this happens where a browser hits us. We see if the page is being cached. If the page is not being cached by us, we have to then go to our internal server and figure out where the real

web server is, go get the information from that server – the index or HTML or whatever – and parse it back through. So there's an entire proxying of the chain of things going on here. There's layers and layers of caching happening here.

Now, when you throw in DNSSEC here, we have to be on both sides of it. We have to be providing DNSSEC records – things that are signed – but also receiving IP addresses from DNS queries that we're doing to the origins and ensuring that they are correct because the other side of our business is to be secure. We have to verify signatures on the other side, as well. We'll be doing both all the time.

This becomes particularly important because people use CNAMEs with us, so what we'll see is things changing fairly rapidly, but we need to look up CNAMEs. We're going to get it from both sides with DNSSEC because of the need to crypto with DNS.

Yeah. So as I said, we deal with DDoS constantly. A quite typical sort of attack for us is 50 million packets per second against a single name server. Those happen all the time.

There are large gigabit per second attacks that are just sort of [inaudible] to try to take out our links. The 50 million packets per second things we actually want to drop and filter from it the legitimate traffic, because when it was reflection attacks, it was actually relatively easy to deal with because you could figure out where the reflections where coming from and start to make decisions about what you know.

What we're now seeing is what is apparently legitimate traffic at very high data rates, and within that there is real traffic. Typically, what

we've seen is, when an attack happens, it's something like one in a million to one in three million packets are legitimate. We've done a bunch of work – I can tell other people about it – to actually filter out nonsense. There are a variety of different sorts of attacks.

When we do DNSSEC, again, we're going to want to reply successfully to the people who are asking us for legitimate queries, even though we've got this very high data rate.

One of the things we did within our DNS server was to deal with this attack information. If you look at the way in which some external DNS is done to deal with attacks, it's good if you are a recursor and you're receiving requests from people and you might receive too many of them.

For us, because of the data rates, those techniques don't tend to work, so we have relied quite heavily on statistical analysis in real time to identify attacks as they occur and what's a legitimate – some simple things. Obviously string matching, because sometimes we'll see exactly the same query, which is bogus, being made over and over again. That's fairly easy to do. Llengths of packets, which is really scary, and also some live regular expressions.

Quite recently, what we've been doing is using the Berkeley Packet Filter. We have code that will dynamically actually create Berkeley Packet Filter code and push it down into the NICs that are running. In our infrastructure, I actually do the packet filtering there, and that's been quite successful. That's been written up in our blog if people are interested in knowing the technical details. Basically with IP tables with BPF you can do a lot.

We have another special feature, which is CNAME flattening. What this does is if somebody asks us for an A record, rather than giving them a CNAME and, "Here you go. Here's all the details," we will internally follow the chain of CNAMEs and synthesize an A record. So we'll say, "Here you go. This is what this is." And we'll do that at the apex as well, if people ask us to.

There's quite a lot of details in how to do that. Basically, you can never, ever tell someone, "This really was a CNAME." Otherwise, recursors will get very, very unhappy about what's happening.

Again, this was done for speed, so there's a single query, and here's the A record, and go get it. With DNSSEC, this has some challenges because of course now we're internally following a chain of things, but we're going to need to verify that the chain is valid, and also that we're going to sign the thing that we end up coming up with, right, because we will have synthesize the record say this exists. However, this is a feature that is very widely used.

I think I talked a little about this. What do we do to solve DDoS? Anycast is key. We Anycast everything, both HTTP and DNS around the world. Depending on four customers who are themselves DNS providers, we will proxy DNS for them and do the filtering of attacks as they happen.

So we wrote this thing in Go. Just to say very briefly about the experience with this because it's quite a large thing to write an authoritative DNS sever and I've got even grayer than I was at the beginning of it, what we found is that Go has proved to be extremely reliable and high performance for this.

We started out with it with some other things and perhaps traditionally would have written this in C, but it turns out that doing it in Go is great. We have machines with a lot of cores, and we have no problems with the performance of this for the core DNS serving at our scale.

When it comes to the filtering, we like to things either in the kernel, or if we can avoid it, not in the kernel at all, but do everything in Userland because the kernel switch is becoming expensive, as actually are IRQ storms at 50 million packets per second for a single machine. You have some interesting challenges there.

So it's essentially a reverse proxy for DNS. There is actually a real authoritative server behind it, but you wouldn't know it because RRDNS is doing all the work. It does a lot things: the filtering I talked about, caching and low-balancing across the machines internally, as well as the CNAME flattening. The point is this thing just looks like it's an authoritative server that allows us to deal with the sorts of attacks and the sorts of traffic we deal with.

This almost reads like a sales slide, but the point is for us is because of our scale, people are changing DNS records all the time, and as the attacks are happening, we're changing IP addresses all the time. So it's extremely dynamic. It's not a file that we've generated and we can sign everything in it and just sit there. It's constantly changing. So that was one of the reasons why we worked on the server.

Let me skip over this because I don't think this is super, super exciting.

The big thing in the filtering is we try to do everything outside of the server, but the stuff that happens inside the server is mostly statistical.

We use the heavy hitter algorithm to look at IP addresses that are hitting us hard or query types that are hitting us hard. Quite recently, actually, in that area, what we've seen is interesting combinations and flags inside DNS packets in attacks. The actual domain names themselves were almost irrelevant. The signal to filter was the way in which they set certain flags. I'm happy to share the actual details with people. Most of this stuff is happening automatically. We will try to figure out what a pattern looks like for dropping traffic.

So, DNSSEC. We are going to provide DNSSEC services, both for the people whose records we actually mange ourselves, which in some ways is relatively easy because we can sign those records. Obviously, we're going to have to put the key signing key with their registrar, but after that we can control the zone signing keys. But we're also going to do it for the people we proxy DNS for. So there are some large DNS providers who are actually hidden behind Cloudflare, so then we have to deal with how do we proxy that traffic.

The most obvious thing is just to be a clear proxy, so the upstream provider actually signs the records themselves and we just take the traffic and pass it on. So we do all the filtering we would normally do, and they're responsible for the actual records themselves and also dealing with the issues around the ability to enumerate the zone. So we'd push the whole work onto them.

We can offline sign things that we need to sign, and we just push it away. Now, the problem with this is that we can't do CNAME flattening because now there's no computation. So the people who do this would lose that particular feature.

A big question for us – and I think the answer honestly is yes – is, "Should we be validating the signatures from the upstream if we're proxying?" if we're just passing on the traffic and having done filtering of attacks, should we at the same validate on every request that the signature is correct? That's quite an expensive proposition if we do do it.

Part of these slides are I'm interested actually to get feedback from people on if they were using some of this, what they would expect us to do, because we're building this out, and understanding what the concerns are around this sort of service would be interesting.

Obviously zone enumeration is a problem that people talked about all the time. Cloudflare has quite a strong privacy thread throughout everything it offers, and so we don't want people to be able to enumerate zones. This is particularly the case because there are people who will use us for protection against attacks, but they will leave somewhere in their DNS records some addresses, some names, which point directly to their own IPs, and we're not actually protecting those IPs.

Now, that's not actually a very good idea because obviously we can't protect them from attack, but you'll find people will have some sort of secret names they would rather not were out there in public. So enumeration is a big problem for us because with that we'd be lost completely. There's security through obscurity what people are doing, but when you've got a million people using your DNS, there are actually quite a lot who are relying on that sort of obscurity.

Obviously, these things are well known. These numeration and dictionary attacks can be done against these things, so we want both [inaudible] and we want to go to do CNAME flattening.

So what we're going to do is live signing. As requests come in, we're actually going to sign them so this doesn't happen. And it also allows us to lie. We can do DNSSEC white lies so we can stop people from doing enumeration.

Live singing is something we can do, but we can't do it at 50 million packets per second, so the core issue we're going to face is what happens when we get attacked. The big problem is that there is a hysteresis where the attack starts, and then you don't necessarily know it's an attack. It could be that someone's turned on some recursor somewhere. Actually, even as the sun moves around the world, you'll see the DNS traffic move as people wake up and the recursor starts asking for things. So you get these floods of requests and we're going to have to sign them.

The current approach is that we're going to have a hybrid approach where we will have a pair of zone signing keys. The reason that we are concerned about this is we would rather limit the exposure of keys to our end machines as much as possible. What I mean by that is we are very concerned about what would happen if a machine was compromised in our network, one of the ones that's on the edge that people actually contact. We would rather not be keys on that machine, or we would rather there were as limited number of keys on that machine as possible.

So the plan is to offline sign the records and the source of requests that we think we're going to get, but when an attack starts, distribute a newly-created key, ZSK, to the machine so they can do live signing, and also they can lie. So obviously those keys need to be signed by the key signing key, but that allows us to, when that happens, start signing things on the edge so that we can try not to have a gap while we're dealing with figuring out what the attack looks like. We're probably talking in the order of seconds that these keys have to live, but they have to be out there.

That means that under an attack, what we think we're going to do is lose the ability to do CNAME flattening, and also we'll probably not be able to lie so effectively under an attack. This is a weakness because of the cost of attacks, where if you wanted to enumerate us, you'd probably attack us first and cause us to change keys.

So we are very interested in solutions about how we can do this because what we would much rather do is live sign everything so we get these problems to go away. But it's not going to be possible under very high attack rates.

As I said, we offer this as a service. We think that a lot of people are going to want to do this because they're not going to want to get up DNSSEC themselves. We've seen with DNS providers for whom we proxy that this is a big, complicated and scary thing for them to do, and they'd like us to do it for them. So in this case, we'll actually do singing for those folks. So we won't be proxying the DNSSEC. We'll be proxying DNS, and then turning it into signed records for those people. Again, we're interested in, especially people who run DNS, what their concerns

would be about that, what they want to have in such a service, because as we build it, we're trying to do things in a secure fashion.

Obviously the simplest solution is live, so we sign everything. So do live and then we push out keys to the edges. Now, there are a number of things we do with keys. On SSL, we have a thing called keyless SLL. Keyless SLL, the private key, is not stored on our edge machines. It's actually stored on a separate machine and there's a secure protocol for performing cryptographic operations on the private keys which are signing, or decrypting the master secret in SSL.

We will do something similar for the zone signing keys for DNSSEC, and we're actually building our infrastructure using TPMs on those machines so we can ensure that the software that's running on there is our software because we want to be sure that these keys don't ever leak out somewhere.

This will allow us to do CNAME flattening and to do NSEC3 white lies on the edge. This is where we're pushing things. Then we'll have a shared cache so we can cache things on the edge. It's not just possible, because of course caching is a large part of our business.

I think the last talk about registrar stuff was interesting because what we'll do here is we will want to integrate with registrars so that we can work on the key signing keys, work with the key signing keys that people have with the registrars, and update them if we need to in a sort of API fashion as much as we possibly can. So obviously talking with registrars about the best way to do that is interesting for us so that we can provide this.

Ultimately what happens is the authoritative servers, whether they're ours or somebody else's gets DDoS protection because that's really our business, and acceleration by us doing Anycast around the world for DNS and caching of things.

We're hoping to make it work with existing DNS providers who have DNSSEC by proxying, or if you don't have it, then we'll be the DNSSEC for that.

On the last one, we're certainly interested to talk to people about how they would want to do the key signing keys in that case. Who would own them and how would they get updated? Because that's a non-obvious issue. We could just create them and be that person, but then we're talking to the registrar, and there may be issues there.

So DNSSEC adds an additional layer of complexity to DNS, which is no surprise, but it's particularly hard when there are large attacks against DNS infrastructure.

I guess that last thing is kind of sales-y. "Let us do DNSSEC for you." There you go. I'm not a salesperson, but I'll try to fake it if you want. All right. I think that's the last slide, isn't it?

CATH GOULDING: Yeah, it is. Thank you very much, John. We have time for some questions. Please go ahead and be sure to state your name.

ROLAND VAN RIJSWIJK DEIJ: Roland van Rijswijk-Deij, SURFnet. You mentioned that you're going to have two zone signing keys, and I'm wondering what rollover strategy, if

you're going to rollover, and what rollover strategy you're going to be using for those because that would impact the number of keys that's going to be in the DNS key query result, and that in itself could be a very interesting query to use for DDoS attacks. So you might be shooting yourself in the foot quite royally. I'm assuming you thought of this, and I'm wondering what you're going to do.

JOHN GRAHAM-CUMMING: The honest answer is I don't know the answer because I'm not the crypto guy who thought that out. He was unfortunately going to be here and couldn't be here. I will find out because we are terrified of being a DDoS provider.

UNIDENTIFIED MALE: For obvious reasons. You'd be generating your own business. I'm sorry. That was cruel. Sorry.

JOHN GRAHAM-CUMMING: No, no, no. That's fine. I used to do anti-virus stuff. Of course, in the anti-virus world, everyone thinks that the anti-virus people write viruses, so it's the same issue. It's terrifying. We actually do some stuff with ChaosNet. You can query us using Chaos on our servers for random craziness, and there is a stupid thing in there where if you query us and ask for our logo, we will give you our logo in ASCII.

When we did it the person who designed the code has a little thing saying, "We will only do this over TCP because this could be…" Can you

imagine the Cloudflare DDoS where we send the logo to people as the DDoS? That would be unfortunate.

So no, absolutely, I'll find out for you, and if you want to grab me afterwards, it would be interesting to talk with the crypto guys.

UNIDENTIFIED MALE: All right. Will do. Yeah. Yeah.

CATH GOULDING: We have Dan York.

DAN YORK: I want to ask maybe the obvious question of, "So, when?"

JOHN GRAHAM-CUMMING: Yes. This year. Definitely this year. There's already work going on on this. Our DNS, our server, has EDNS code, but it hasn't been rolled out. That needs to be done. I think that all of the hard thinking has been done on DNSSEC by the crypto folks within the company, and is now simply a matter of programming.

So I would say I don't know the exact thing. I believe it's the beginning of Q4 we plan DNSSEC.

DAN YORK: Well I just want to say personally it's great that you're here because I'm glad to hear what you're doing in that space because we've certainly seen that the CDN space is one that is a blockage for a lot people in

getting their domains signed because they come down to needing the CDN to provide it, so they're ready to go and then they wind up with, "Oh, we can't do this."

So I'm glad you're working on this. It's great to see. I look forward to it, and I would have to say that if you're looking for beta testers, you probably got a room full of them here who would be glad to help in any way, shape, or form.

JOHN GRAHAM-CUMMING: That's nice of you to say that. The reason we wanted to talk here is—

UNIDENTIFIED MALE: Cheap labor.

JOHN GRAHAM-CUMMING: No, not cheap labor I think there's no cheap labor in DNS. It's just the people who work on it. But the reason was we are pushing this very hard and it's not obvious what the solutions are, and we're not going to come up with all the answers ourselves. We have ideas about what we want to do within our constraints, but I wanted to say, look, we would like to work with people on how we make this work successfully for the people for whom we're authoritative and for whom people we're proxying and because of this – and there are a couple of other protocols that are not DNS-related – are things that we want to absolutely be on the sort of leading edge of some of these things because that's part of our business, basically is to be up to date.

So yes, please reach out to me and Nick Sullivan, who couldn't be here, as the crypto guy, as well.

DAN YORK: Well, when you guys put the IPv6 on switch, you made a lot of websites easily bumped to IPv6, so personally I'd love it if you'd have a DNSSEC switch like that that would just make that all work for a lot of sites, too.

JOHN GRAHAM-CUMMING: Yeah, I think we will. So what we'll probably do is what we do with the IPv6. Originally with IPv6 it was you had to press a button and we would make your site IPv6, and then at some point that button just got greyed out and was always, so everybody gets IPv6.

I'm sure with DNSSEC what we'll do is we'll say, "Do you want it?" Okay, we'll switch it on. Then at some point we'll say, "You're getting it. Here's how we deal with it." The complexity is with IPv6 it's relatively easy because we can say, "Here's an address." With DNSSEC, there's a question of who's got the key signing key. So with the registrars, it becomes interesting. But yes, we're definitely doing it.

CATH GOULDING: Rick Lamb?

RICK LAMB: Thank you very much for sharing this stuff. With Cloudflare, early on, we were always having problems every time I'd say DNSSEC around your CEO.

JOHN GRAHAM-CUMMING: Just ignore him. What does he know?

RICK LAMB: So this is really cool. Maybe I'll be looking for a job.

JOHN GRAHAM-CUMMING: [inaudible]

RICK LAMB: This is exactly the fun stuff. My serious question – my geeky question – is, so you have a TPM. You're using that to form sort of protected set of encryption. What are you going to use for an HSM, or are you? Are you going to keep this all software?

JOHN GRAHAM-CUMMING: No, it's going to be all in software.

RICK LAMB: Okay. No, that's fine because you've got to be flexible. This has got to change on the fly.

JOHN GRAHAM-CUMMING: Yes, it has changed on the fly, and the important thing about the TPM is it will allow us ensure that what we've distributed is what we've said we've distributed it to an individual machine.

RICK LAMB: Right. So you're not using it as an HSM, because you can use the TPM [inaudible].

JOHN GRAHAM-CUMMING: Yes, we can. So okay. There are some keys that will be in the TPM. Not these keys. There are sort of special keys. Then on top of that, we can then ensure that what's got delivered to the machine is what we want delivered.

But for a lot of things – for example, for a lot of the DNSSEC, we're going to offline sign everything we can and not have the key on the machine at all, just like we're trying to do with SSL as well, which is no keys on that machine.

We would like you be able to walk into a data center and steal one of our machines. We prefer you don't do it, but if you do, we want it to be the case of you're like, "Okay. Great." So you got a copy of our binaries. Well done.

RICK LAMB: Yeah. Well, I hope you keep sharing. This is great stuff. This is very educational.

JOHN GRAHAM-CUMMING: Great. Thank you.

CATH GOULDING: Any other questions?

RYAN DIMBLEBY: Just a quick question on the overhead of implementing DNSSEC in terms of you showed a slide at the beginning—

ROY ARENDS: Can you specify your name?

RYAN DIMBLEBY: Oh, sorry. Ryan Dimbleby from IG. How much slower are you now?

JOHN GRAHAM-CUMMING: We don't know is the answer. Do you mean how much slower do we think our DNSSEC crew is going to get when we sign them? Hopefully no slower because for normal stuff we will have offline signed, so it will be part our – we have a pipeline that occurs where all the stuff that people enter whatever weird stuff they enter in their DNS editor and it turns into this binary stuff that we distribute out into our network and goes out through a thing we use called Kyoto Tycoon. That will get distributed with signatures.

So our pipeline may slow. I would imagine that the issue will be what's the delay between you changing the DNS entry and us actually putting that on our edge servers? That might change with signing.

The real issue about slowness is when an attack happens. What happens when we have to start doing live things? So at the moment, we're not too concerned about this.

RYAN DIMBLEBY: What about in terms of webpage load times before and after?

JOHN GRAHAM-CUMMING: Hopefully, if we're not changing the DNS lookup time much, that won't be an issue. It's definitely something we don't want to cause a problem with because that's our business. So it will be figured out. I don't know what the correct answer is. Maybe it's going to be some hardware. Maybe it's a software thing. It has to be the case that we don't mess up our DNS time. We're number two on that list and we'd like to be number one.

CATH GOULDING: Thank you very much, John. I think we should move on, so please join me in thanking John for a very stimulating talk and presentation.

[applause]

As we switch over to the next panel on HSMs, someone has mentioned that there's someone from HSBC here who might want to say something. I don't want to put you on the spot, but if you have a couple things to say – he's like, "Oh, thank you so much."

UNIDENTIFIED MALE: Yeah, so hi there.

UNIDENTIFIED MALE: Get a little closer.

[NEIL SMITH]: Is that better? Yeah, I guess. So, hi there. I'm [Neil Smith]. I work at HSBC. There was a question asked earlier from somebody around why the adoption rates of DNSSEC isn't being taken up on key transactional

services or high-value services from key brands. The reality is for us with DNSSEC it brings some benefits. We recognize that, or certain brands and certain companies will do.

However the prime refocus for ourselves is not just availability but also risk and that risk mitigation. The challenges that DNSSEC brings, especially across our portfolio of in the domain space as we have it today, is the ability to predict and be consistent with the rollovers of the DS records into those registries.

If it's not standard, consistent and we've got a much larger overhead, not just in a resource perspective for managing that, but also from a risk perspective. If we miss a rollover or if something happens, we're effectively increasing the risk for a limited benefit, and that is the challenge for ourselves.

So if there was a consistent standard process that was very robust in the ability for us to deliver DNSSEC across our portfolio of services, we'd obviously adopt it. We'd love to adopt it. But that is our challenge, and I would ask the registries and those involved if people here can build a consistent process and a standardization across that piece that makes it easy for us to adopt, we'd love to adopt it.

RICK LAMB: First of all, you're saying standard process for uploading I guess DS information across registries and registrars, etc.

[NEIL SMITH]: Correct.

RICK LAMB: Okay. And it's the key rollover part that's the challenge for you right now?

[NEIL SMITH]: Yeah.

RICK LAMB: Okay.

[NEIL SMITH]: And there may be other areas as well. It is purely about availability and risk mitigation, and we're not going to take and implement something that brings us some benefit that may or may not be seen by our customers but gives us a massive potential for risk for if something goes wrong taking out a service which we've seen.

RICK LAMB: No, thank you very much for that. That's consistent with other feedback we've heard, so thank you for being here and saying that from the HSBC perspective because it's really good to hear because that's exactly the kind of thing we've seen that we need to work on to automate that piece of the DS part.

If Patrik Falstrom were here, as he has been at other events, he would talk about the challenges he has a registrar interfacing with all the different registries that use different ways of interacting and different

things and what a massive headache that is right now. So there's definitely some work to be done on there.

One quick question. Do you primarily interact with registrars, or do you actually do some interaction directly with TLD registries?

[NEIL SMITH]: We have a portfolio management company that does that for us.

RICK LAMB: Okay.

[NEIL SMITH]: The one place where we may see that change is in our own gTLD where we'd be forced to maybe do that, but we'll have the opportunity. We'll only have one key to potentially roll over if we automate our own management inside.

So that may bring it within our own space, but across our portfolio as soon as that's available.

RICK LAMB: Great. So let's exchange some contact information and I think there's others around here too who'd be very interested to talk more.

[NEIL SMITH]: Okay.

RICK LAMB: Thank you for that.

JULIE HEDLUND: Yeah, thank you, and sorry to put you on the spot. At any rate, than you very much for that. We'll go ahead and move and I'm going to turn things over to Roy Arends from Nominet.

ROY ARENDS: Thank you, Julie. We're going to have a panel about hardware security modules, their benefits and challenges. I've got three of my peers here, and we have Rick Lamb from ICANN, we have Mark Southam from Ultra Electronics AEP, and we have Roland van Rijswijk-Deij from SURFnet. Each panelist will get ten minutes to present their slides and to talk about what they do.

I would like to ask you to keep your questions until after the 30 minutes of presentation, not during nor after an individual person presentation, but just after the presentations. Thank you.

First up, we're going to have Richard Lamb from ICANN to discuss Hardware Security Modules (HSMs), benefits and challenges. Thank you.

RICK LAMB: Okay. Well thank you very much, Roy. Can you be my finger? Okay. All right, next slide, please. Everyone knows who I am.

I do a lot of training. One of the things I get all the time is, "These hardware security modules. We're going to deploy DNSSEC. Do we have to have one of these things?" We all like cool toys and things that look

really high-security, so there tends to be a leaning toward automatically saying, “I need to get an HSM.” There are a whole bunch of them up there. I can go through them. They’re all interesting.

But you’ll notice up there you also have a smart card and a regular USB fob. That is just as good as an HSM as possible. The chip on the right hand up there is a TPN chip built into many Dell and other servers. The rest of those things you’ve probably seen before, but that pretty much captures the main ones you see out there now. Let’s go to the next slide, please.

You really have to ask yourself, “What are you protecting?” I know this is a presentation about HSMs and we like them, but I’m saying you really have to ask the question. Ask yourself, “What are you doing here?” because in an awful lot of installations I’ve seen, the DNS operations themselves, or the zone file maintenance or whatever you want to call [inaudible] is not that procedurally precise, shall we say. So to have all this in there and have the data itself? Garbage in, garbage out. That’s what it is. It’s a problem. Next slide, please.

A lot of times, this is all you need: a [inaudible] bag. I always carry one with me all the time. I think it’s always important to have. You never know when you have to transfer cash or something. But they are just cash bags. This is an amazing amount of security you actually get from this.

So something like this: a flash drive. I know a couple installations. They have a flash drive. They have the KSK on there, ZSK in there. Multiple people have to be involved in order to get at that. I’ve seen places

where they have an HSM there and one person can access everything. So what's the point of that as well?

So a lot of times it is this that you need to focus on: access control, reasonable safe – hotel safe, anything – in place. Next slide, please.

But this is usually the problem. You guys all know this. There's bad passwords and stuff but no documentation. As engineers, we hate documentation. We hate doing this. But our friends at .se – that's up there on purpose – they really blazed the trail here by offering creative comments, licensed DPS that unfortunately I've seen a couple places where they've just copied it, including the translation mistakes from Swedish to English. You know who you are. Come one. Anyway, nonetheless, a very good starting point, trying to do a DPS and following that. Next slide, please.

And sometimes it's this, at least in two places I've seen – well, no, they didn't use that random number generator. You've all seen this. That's why I put it up. I don't have to explain it. But they'll have a laptop or something like that in a safe, which is great. It gets pulled out twice a year. Not much entropy in that thing, and they're just pulling off a dev/random. So it's probably fine, but that's something you should consider – your random number generator. Next slide, please.

This is really what you need to look at. What are you protecting? You've got to ask yourself this question. Who's your customer? Your customer may actually want you to have the cool bells and whistles and iris scanners and all this stuff out there, and whatever. I'm the guy who worked on designing the root KSK management system, and I'll be honest with you, put a lot of stuff in there. But who is our customer?

Our customer is a bunch of people that don't trust ICANN. Well, maybe that's everybody. But we had a very high bar to cross, so we had to really take every precaution and put everything in there.

But that might not be your case. I'm saying don't necessarily either use – have an HSM excuse not to deploy DNSSEC or to automatically go there and waste your time.

Setting expectations is one of the most key things here. It's kind of tied in with who's your customer. Who are you selling to? Who's the one that's going to care about this stuff? You need to understand from them what they expect from you and what you can deliver. Cost actually is not that big a deal at the end. Next slide, please.

This is a great slide from my colleague Phil Regnauld from Network Startup Resource Center. If you look at the overall picture of what DNSSEC protects, it really doesn't protect everything. So you really need to look at some of the boxes that are not green, as well. I think all you guys understand this: the database file itself, access to slaves, etc. Next slide, please.

Now let's say we're going to get an HSM. Fine. You really want to look for some common standards. PKCS11 is wonderful. My colleague here, Mark from AEP, they were very instrumental in helping us try to understand PKCS11. And you want to go for something with PKCS11, but also be prepared that that is not a fixed standard. There are a lot of options. OpenDNSSEC, for example, does a wonderful job of trying to hide all these differences, but you can pull your hair out trying to get this stuff, obviously, to line up because everyone's got it a little bit different.

But basically, all you're looking for is these two function calls, C_Sign(), C_GenerateKeyPair(). That's it. Avoids vendor lock-in. That's great. I've heard of something called KMIP. I don't know how many HSMs supply this, but one of the things you might be interested in if you're going to go the HSM route is how do I export a key from one vendor and put it into another vendor? That is a very hard problem because that part of the management of HSMs tends to be vendor-specific. So you might want to look at something like that. Next slide, please. I think everyone can see that.

Certifications are important. The other thing people talk about HSMs all the time is it FIPs Level 3, 4? Is it common criteria, which seems to be something. I'll leave that to Mark to talk about a little bit. Various things, but this is up to what government you're under. Brazil has its own, for example, which is cool. A lot of links you can look up later. Next slide, please.

Smartcards/tokens. Japan uses a smartcard. I think it's a wonderful approach for just the KSK. It's not something that changes often. There are a bunch of other solutions like that. That CardConnect thing there – I have a couple of them in my backpack. That's really good. They actually make something called the SmarCard HSM. It's about $20 U.S., I think about 16 or something eu. It's wonderful. It has a way to export the key from card to card and make backups. This is a relatively thing that's good.

TPM is built into many PCs. Very messy API. I would suggest not going down there. Costa Rica was very courageous and brave to go down

there and they have something that works completely off of that. Two minutes? Okay.

Tokens, open source PKCS11 drivers, and random number generators are very important. But still, smartcards are going to go like one to ten. Next slide, please.

I think this is the last slide, probably, so you got some time here. The random number generator is something that we all really do have to keep an eye on and look at, particularly with the Snowden revelations and all of that. It is something that you really do want to consider. A lot of the people that I talk to want basically the cover-your-ass slide, so they want to make sure even the random number generator is certified.

Your mileage may vary there. You may actually say, "No, I really want a random number generator," but whatever. You can follow standards. That lava lamp was one of the very early really good random number generators. Someone had a camera pointing at that. It's called LavaRand. They took the last few bits out of that. Very good. That little fob in the lower right hand corner there, well that's true. You've probably heard about it, right? It's kind of funny. I love lava lamps. I do. I'm that age. But they have that little thing in the fob [inaudible] corner, I think that's made by a small company in the UK, actually: [Syntac]. They make something that's pretty good. There's another made by the [Finns] that's been around for a while.

I'm mentioning these different things because sometimes, given the environment we live in now, your customers may say, "We want something not made in the U.S.," or whatever.

The ones built into the CPU chips – I think many you've heard te FreeBSD guys have said, "We're not going use our random number generators built in there now." But again this is all policy and up to you.

Next slide. I think I'm done. That's it. That's it for me. I think I made it in time. Thanks, Roy.

ROY ARENDS: You indeed have four seconds left. Let's just check that for a second if this works. Yeah, wow. Okay, so next up on my list we have Mark Southam from Ultra Electronics AEP. Mark, you have ten minutes. Thank you.

MARK SOUTHAM: Hello. Thank you for having me. I'm going to just talk about HSMs from being in a security company – a secure point of view. Obviously, not everybody's going to choose certified HSMs – shall I be more specific? – for certain types of data. In the case of DNSSEC, obviously we're protecting DNS data. Can I have the first slide?

Certified HSMs are designed to standards that have been developed in the U.S., and when you're looking at the NIST standards, and then globally – or we're trying to be global – when you're talking about common criteria.

When you think about securing a solution, a well-known strategy is building layers of defense. What HSMs provide for you – and certainly certified security HTMs – is extra layers of defense for that particularly sensitive piece of data, cryptographic keys.

So while you might normally have the usual layers will be firewalls, you're creating different layers in your network, and it's all to protect user data, ultimately. Sometimes you'll use cryptographic keys for part of that. You will use them for remotely accessing net data, for authenticating to gain access to that data if you're using two-factor authentication. You're using it for your web service, presenting that data to your customers on the Internet.

HSMs provide a mean of adding all the extra layers of defense on the bottom to any of those applications, so I'd like to think as an HSM as not something you're using for a particular application in your network. It's something you're introducing as part of your security ecosystem and using it across your organization. So not just DNSSEC, but all of those things I've previously mentioned.

The kind of layers you're getting are physical separation, which is something an earlier speaker talked about. He didn't want any keys on his servers on the edge. So if we take it as a given, like some [inaudible] I've spoken to that anybody's on your network, you build up these layers to make it harder to get to the things they really want to get to. That's kind of the thing HSMs are trying to do for cryptographic keys. So there's physical separation.

The hardware random number generators that I work with certainly have quality random number generators – sorry, HSMs I mean have quality random number generators – and that's obviously a crucial part of cryptography, the seed that feeds the algorithms. The more random it is, the better the jumble of numbers that you get out ultimately [inaudible] text.

There's different levels of certification that you can get. As Rick implied, Level 3, Level 4 in the sort of NIST standards. There's also different levels in the common criteria standards as well. Both are currently undergoing some metamorphoses. Snowden's impacted the NIST standards and common criteria has been fading in popularity for some time, so I think they might have agreed already on a new way of doing that in which they'll have common protection profiles for different types of devices.

So ultimately, they will have one for an HSM, and you'll have a better feel when comparing HSMs that they are better than one another, so to speak.

So you've got different levels of certification. That's going to give you a better feel about the levels of security that it's applying to your solution. You're getting multi-factorial authentication, so removing one person's access to the keys, as Rick already implied earlier. One person access is a bad idea. So HSMs tend to provide quorums, and some HSMs provide quorums just to turn their services online. Some take it a step further and allow you to enable and disable signing, key generation, and such.

Finally, the provide levels of security from the application point of view, so PKCS11 generally gives us the mechanism to use pins to enable application access. Okay, that's that slide. Next slide.

Just a bit about certification and cryptography, a little bit more. A good crypto states that a cipher should only been use if it's been thoroughly crypto-analyzed in a global community by the mathematicians, so why not use the same mechanism to choose the tools that you're going to use to perform that crypto?

So any HSMs that have certification have been tested thoroughly by the communities and labs globally for many years. So you're getting that peace of mind when you've chosen that that they've been tried and tested, similar to the crypto-analysis that I talked about earlier.

We've mentioned a bit about common criteria. That's a bit about [inaudible] analysis [key per HSMs], I think is the most secure HSM globally. It has both FIPS140-2 Level 4 and EAL4 plus the advanced vulnerability analysis. So yes, if you're using pertinent to the argument we're having today has any form of cryptography performed using software undergone such analysis, so globally over such long periods of times by labs, paid analysis. Next slide.

Just an example of the kind of attack we're trying to prevent with HSMs. Heartbleed was a nice, recent, massive example, the vulnerability in the software. Always hard to prevent accidents and mistakes by humans who create that software. A bug in Heartbleed gave access to the private keys, and man in the middle attacks all are consequences of that.

So that's me. Thank you.

ROY ARENDS: Thank you, Mark. Next up we have Roland van Rijswijk-Deij from SURFnet, who's going to talk about his project SoftHSM: A Brief Overview. Thank you. You have ten minutes.

ROLAND VAN RIJSWIJK-DEIJ: Okay. Thank you, Roy. Yes, my name is Roland Van Rijswijk-Deij. We take part in the OpenDNSSEC project, and SoftHSM is an effort from that project. Next slide, please.

SoftHSM is a bit of a misnomer because it's actually not an HSM. It's a software implementation, but what it does is that it implements that common API that you see for all the HSMs on the market, which is PKCS11. As I said, it was developed by the OpenDNSSEC project because, as Rick already mentioned, the OpenDNSSEC relies on the PKCS11 interface for all cryptographic operations so that the software is interoperable with all the HSMs that are on the market.

Obviously we needed something to test, so SoftHSM was first devised as a test PKCS11 that allowed us to do rapid development of OpenDNSSEC in the starting phases. But note that we do test with real HSMs as well, because as Rick managed, PKCS11 comes in many flavors.

Another goal is that we provide a free and cheap alternative for users that do not need a real HSM, where the thing that they are protecting is basically not worth investing the kind of money that you need to spend on an HSM, but they do want to do DNSSEC. They do want to use OpenDNSSEC. So we needed an alternative for them. Next slide, please.

So we introduced SoftHSM in 2009. It was developed with the focus on supporting those PKCS11 functions that you need to do DNSSEC signing, and basically it offers very little security at all. So it stores key material in the clear, just like BIND does, or at least used to do. It has a database back end where it stores the key material, and it can import keys from BIND, so you can easily migrate from BIND signing to OpenDNSSEC and vice versa.

We use a cryptographic library called BOTAN underneath. That was specifically chosen because there was – how to put this mildly – lack of trust in open SSL among the developers of OpenDNSSEC, and I think Heartbleed has proven that that lack of trust was warranted. Next slide, please.

What we did was over time we slowly extended the features that are in SoftHSM to also support other scenarios. We saw that there was pickup of SoftHSM outside of the DNSSEC community, and a number of patches were contributed so that we now support a more full implementation of the PKCS11 library, where for instance we're able to store X.509 certificates and support newer signature schemes, such as RSA-PSS.

We are currently at release 1.3.7 as our production release. I also want to tell you a little bit about our next generation of SoftHSM. Next slide, please.

Oh, sorry. It's all open source. Sources are available on GitHub. There's a Tarball available from the OpenDNSSEC website, and we support loads of different platforms, loads of unique [inaudible] and also Windows.

So we are working on our next generation of SoftHSM, which is an effort that we started in 2011, and it's been slowly developing over time. The idea behind that is we did a redesign where we wanted to make it as secure as we can make it in software.

We can never reach the levels of security that you can have with dedicated hardware that you can separate from all the rest of your network and all the rest of your equipment, but we wanted to make it as secure as possible because we saw that most users that use

OpenDNSSEC – at least the crowds outside of the TLDs that use OpenDNSSEC – don't rely on hardware security modules, and we wanted to give them extra security.

So what we do is we now store all sensitive key material in encrypted form. We make sure that it only gets decrypted in memory when needed, and we suppress paging of sensitive data to disk, because obviously if you run an operating system, if it runs out of memory, it's going to start swapping stuff of [to] disk, and you don't want that to happen to your decrypted key material in memory. Next slide, please.

Another issue with the first version of SoftHSM was that it didn't scale very well beyond about 50,000 objects stored in it, and we redesigned SoftHSM Version 2 such that it supports much larger deployments. We also introduced a cryptographic abstraction layer so we could support multiple cryptographic back ends so if there is a bug in something like OpenSSL or in Botan, then it will allow us to slide another cryptographic library underneath and keep SoftHSM running without the users noticing. So the key material stored in the SoftHSM database is completely independent of the cryptographic library that's underneath it.

Currently we support Botan because we supported that in version one, but we also support OpenSSL.

Also, to deal with larger-scale deployments, we now have a file-based storage mechanism that may perform better in some circumstances, but we also still support a database storage model.

From the ground up, we've included all relevant PKCS11 features that you would need in most PKCS11 applications, so not just DNSSEC, but also other use scenarios. And we support the GOST algorithm by request from some of the users of OpenDNSSEC. Next slide, please.

So SoftHSM Version 2 is still under development. We do believe that it's feature complete. We encourage everybody who wants to play around with it to download the alpha release and have a go at it.

Again, it's fully open source. The sources are available on GitHub. But I do like to mention that we could use some help with further development because we have a limited amount of funding available for the developers that work on this, which is one of the reasons why we're still slowly progressing towards the release.

We do hope to have a production release by the end of this year. We do have a commitment for funding to get that done.

I think that's it. And for questions or comments, come up to me in the break or whatever. Do have a look at our website and do play around with it. Thank you.

ROY ARENDS: Perfect. Thank you, all of you. Roland, nice within time. I'm going to use the remaining three minutes that Roland has to give a shout out to two pioneers in DNSSEC and HSMs.

One of those two I want to give a shout out to is Jakob Schlyter. In 2005, Jakob was one of the very, very first who was able to have a DNSSEC

thing talk to an HSM thing, and his implementation made use of basically a cryptographic smartcard.

The other person is actually here in the room. His name is John Dickinson. He works for Sinodun. In 2005/2006, he was the very first that I know of who was able to join a DNSSEC module – I think it was at time BIND.

UNIDENTIFIED MALE: [inaudible]

ROY ARENDS: No, no, no. Rick made the implementation within BIND, but long before that –

UNIDENTIFIED MALE: [inaudible] DNS.

ROY ARENDS: Oh, DNS. Sorry. But long before that, John was able to use LDNS, which is a library from NLNet Labs, to work with an actual HSM, and I think it was the SCA6000 card from Sun. This was 2005-2006, and I'm very, very glad to see things have progressed this far that we now have actual vendors in the room to discuss this. In 2005-2006, vendors didn't even know what DNSSEC was. It's very good to have you here, John. Thank you.

With that, I want to open the floor for questions, comments, or remarks. I want you to state your name first and then to ask your

question. If the question is for the entire panel, say so. If it's directed to one of our members of our panel, please direct it to him. If there are no questions, I have a few, so let's go for it. Thank you.

So we have John Dickinson of Sinodun first. John?

JOHN DICKINSON: Hi. Obviously I've been doing this for a little bit of time, and one of the big things that I found about HSMs that hasn't been mentioned today – and this applies equally to the SoftHSM – is that in addition to doing security, it's also very good at allowing you to organize and maintain your keys.

BIND, or in the old days certainly we just spat [K files] [inaudible] disk and they were files with names that had no meaningful use at all, and you had to look inside them to find out what they actually contained. People were doing things like creating directories or using [inaudible] and things like that to lock down their BIND keys.

If you're doing that, you're building a SoftHSM, and in which case you might as well use one somebody else as already written. So even if you don't really care about the added security, it might be worth getting an HSM or SoftHSM just to help you organize and maintain your keys.

ROY ARENDS: Thank you, John. Anyone else with a question? We have three people. First up, thank you.

MARC SIDEN: I have a rather trivial question about OpenHSM. I assume you're careful about zeroization in the code?

ROLAND VAN RIJSWIJK-DEIJ: Yes. I didn't mention it, but you can look it up on GitHub actually if you want to, but yes we are.

ROY ARENDS: Thank you. We have another question over here.

UNIDENTIFIED MALE: [inaudible], .cz. I have a question more about support for signing software in those HSMs. You mentioned OpenDNSSEC. However, as far as I know, this is the only solution that comes back. Recently, BIND 9.10 I think has some PKSC11 support, but have you tested BIND 9.10 against SoftHSM or against Keyper?

ROLAND VAN RIJSWIJK-DEIJ: I'm going to answer for SoftHSM. BIND – I know that the folks from ISC actually use SoftHSM for testing, so I'm assuming they tested BIND 9.10 against SoftHSM. But that's an assumption. You'd have to ask the folks from ISC. But I do know that there is actually someone from the ISC who is contributing to SoftHSM Version 2, so I'd be highly surprised if they didn't test [with] it.

MARK SOUTHAM: From the HSM point of view, I know that [inaudible] support the BIND PKCS11 interface. So they generally work very much – even though Rick

implies otherwise – they work very much the same on the application side of PKCS11. It's just the differences will be the way they translate it into the different hardwares. So I don't think there's any particular reasons why all of the HSMs wouldn't support BIND 9.10 and PKCS11 because they're all pretty much designed around PKCS11 to perform like a PKCS11 device.

UNIDENTIFIED MALE: Just to finish that off, [inaudible] also supports it as well, but you're right. That's a good question. We need more software platforms supporting that wherever they are, whether it's BIND, Knot, or what have you. It's a good question. Not a lot of software natively supports this stuff, or not easily.

ROY ARENDS: Thanks. Next up we have [inaudible].

UNIDENTIIFED MALE: [inaudible]. I'm glad to hear that SoftHSM is getting to Version 2 because I have actually [ports waiting] on the FreeBSD on the moment it really gets stable, so if there's a more updated release, I will – actually [inaudible] release it next week if people want.

UNIDENTIFIED MALE: No pressure.

ROY ARENDS: Thank you, [inaudible]. Anyone else from the floor? Perfect.

SARA DICKINSON: Sara Dickinson, Sinodun. I was wondering if you could make a few comments possibly about the dependency on the vendors that it introduces when people decide to choose a particular HSM. I'm thinking of certain cards that might have changed their pricing models in the past, or ideas about relying on the vendors to provide software updates and important maintenance. Some comments directed at that.

ROY ARENDS: I'd like Roland to answer first, please.

ROLAND VAN RIJSWIJK-DEIJ: I'm going to do full disclosure at the same time. Actually, we at SURFnet don't eat our own dog food because we use an HSM from SafeNet to store our key material for DNSSEC signing, and of course we use SoftHSM in other projects, but we ourselves use an HSM.

One of the things that you need to keep in mind with vendor lock-in, I'm not as PKCS11 where DNSSEC is concerned as Rick is. I think that the common functionality you need for that is pretty much the same among all HSMs.

The trouble is you really need to take care how you design your key storage if you want to be really vendor independent. One of the things you will want to do if they don't support this KMIT thing is that you need to make sure that your keys can be exported under the right conditions so that you can migrate them to another HSM. As long as you take care of that, then the standards are there to be vendor

independent. But this requires detailed technical knowledge about PKSC11, which is one of the main drawbacks.

ROY ARENDS: Are you still FIPS 140-2 compliant if you are able to export your keys in PKSC11?

ROLAND VAN RIJSWIJK-DEIJ: That's a very good question, and the answer is no.

ROY ARENDS: Exactly.

ROLAND VAN RIJSWIJK-DEIJ: I want to follow up on that. That's taking a very limited view of the security certifications because just because your keys have become exportable doesn't mean all the other benefits that those certifications bring don't still apply. So that's a decision that an individual user should make, and even though it may invalidate your certification, you should really ask your vendor, "Does it change anything in the way the keys are treated in the hardware?" Then they should be able to tell you, "No, it doesn't," and then who cares about FIPS?

ROY ARENDS: Exactly. I'm just the devil's advocate here. Mark?

UNIDENTIFIED MALE: Yeah, do you want to say something to that?

MARK SOUTHAM: Yes. Roland's absolutely right. The catch is that security is enabled by default, and security in this case means export is disabled when you first generate a key. So if you want to think about that up front, you need to make your generating keys with the export flag disabled. Then you can protect yourself – well, certainly in the case of mine; I can't speak for the others – but you can protect yourself from disabling global key export until you absolutely need it. You really want to export your keys or you want to perform backups and you can perform ceremonies to do that. But yes, it is possible.

RICK LAMB: Correct me if I'm wrong, or anybody please correct me if I'm wrong. My understanding is if you do the [CNWrap] and you follow all the right protocols that you can't maintain with FIPS certification by exporting and importing keys.

UNIDENTIFIED MALE: I believe so, yes.

RICK LAMB: But as you said, you have to really get to really know the format of the internal smartcard structures. It is not easy. Vendors do want to – well, for obvious reasons, probably – do not want to make it really easy to be able move from one to another.

UNIDENTIFIED MALE: [inaudible]

ROY ARENDS: Hold on. Hold on. First, Paul has something about this to say as well.

PAUL HOFFMAN: FIPS 140 certification is on a device, not on a use of a device. So if the device has a FIPS 140 certification, you can still do whatever you want, such as export. So you are not breaking the FIPS 140 certification by exporting.

But more significantly, which is exactly what Rick has said, there are ways of doing things that will also be protected, and that's exactly where PKCS11 falls to the floor on interoperability is on keywrap once you're moving things – and there's other places where it falls on the floor as well. So the interoperability goes to hell when you're doing things securely.

ROY ARENDS: Thank you, Paul, for keeping us honest. Go for it. You had a comment?

UNIDENTIFIED MALE: [inaudible]

ROY ARENDS: Oh, perfect. Do we have more questions from the room? John has one more thing to say.

JOHN DICKINSON: Hi again. The other issue across early on using these various HSMs – and I did use several of them – was that the quality of the documentation and the quality of the support teams the companies provided varied enormously. I don't know if the distinction is as big these days as it was, but if you wanted for example to do something like create an RSA key, almost all of them didn't tell you how to do that in the documentation because they expect you to use these things as SSLN points, not as DNSSEC key storage.

If you are thinking of getting one, get hold of the documentation. See if tells you the simple things like how to get a key. Ring up the support line and ask them, "Do you support PKCS11?" and see if they actually know what you're talking about. I think this is important as price, really, in deciding on what HSM to get.

RICK LAMB: If I may just say one thing, that's absolutely right. I'll just be honest, one of the reasons we went with using [AAP] for the root was I called all these other vendors up. They were the only ones that said, "Sure. Here's a code snippet on how to do this." No one would help like that.

But you're absolutely right. The documentation has all these assumptions that you're just plugging this into the system and it's very hard to understand.

UNIDENTIFIED MALE: [inaudible]

ROY ARENDS: Sara, you had another question?

MARK SOUTHAM: May I quickly add that crypto engineers are expensive and their time is valuable?

SARA DICKINSON: It wasn't a question. It was a comment. It was that if you go to the OpenDNSSEC website, I can send a direct link out, but Jakob actually put together a comparison of all the HSMs in terms of functionality and performance, and I think that's quite a useful document to read as an introduction to the variety that's out there.

ROY ARENDS: Thank you. Also for the remote participants, you have the ability to ask questions. We see them here so we can ask them for you to the panel. Thank you.

UNIDENTIFIED MALE: I wanted to plug the HSM Buyers Guide as well, that we OpenDNSSEC, which will tell you what you need to look for if you're deciding on an HSM.

ROY ARENDS: Okay. Good. Thank you. I've got a question if no one else has one. Okay. Oh, we have a question from Luis?

LUIS ESPINOZA: I worked in the past with Richard Lamb on the TPM thing. It's really complex. Well, my point is if you want to deploy DNSSEC using HSM, the providers of the HSM provide that and I think that has no barriers. Right now we have the price barrier and the complicity barrier. I think everything should be smooth if you want to go further with this implementation.

ROY ARENDS: I agree. Thank you. I've got a question for Mark from AEP Networks. Mark was the one who had on his slides with an HSM, a Heartbleed wouldn't happen. Naturally, HSMs have software, too. It's embedded in these devices. HSMs need to have device drivers, etc. We have currently a lot of pressure to roll over the root key. The root key is using AEP hardware in order to roll it over. The root key is live the KSK almost five years or maybe over that. AEP guarantees that the batteries will last for five years.

What happens if the batteries run out? Is it easy to replace the batteries? Is it easy to upgrade the software with and HSM? Do we only have to rely on a specific vendor who then needs to touch the HSM keys, etc. etc.? Can you talk to us a little bit about that?

MARK SOUTHAM: Yes. Being a device that's secure in nature, the keys are signed using our own HSMs – sorry, the firmware I mean. Nobody has access to those except our factories. You can only have the devices re-commissioned in our secure facilities.

So that's the nature of the beast. If you want that level of security, you've got to go through a bit of pain with the hardware support. But it's nothing that nobody hasn't got through, and as long as your backups are fine, your key material is fine and you're recoverable.

ROY ARENDS: Perfect. So there's no real pressure in terms of the specific hardware to roll the key because of the five-year lifetime etc., etc.?

MARK SOUTHAM: The five years is a guideline but it's based on whether the device is online all the time or in the shelf. If you're coming up to five years, then it is best practice to change it at that point because you're going to be running on battery presumably when it's in the safe.

ROY ARENDS: Yeah. Okay, thank you all very, very much. I end the questions hereby. I would like to thank Mark, Rick and Roland for their participation on this panel, and for everyone for asking questions. Thank you.

JULIE HEDLUND: And thank you also to you, Roy, for running such a good session. So now our next speaker with be Haya Shulman, and I'll ask her to come up and we'll get started shortly, as we do want to end on time for the lunch break.

[DAN YORK]: While Haya is getting set up, I will just say that we all owe Julie a huge amount of thanks for a number of reasons, partly that she's just awesome in helping organize this and keep it going, but also that she came in last night and realized there were no tables for anyone to eat and was able to work with the hotel staff to get some tables and such in here. So the fact that you will have somewhere to eat today is largely owed to Julie, so thank you, Julie.

JULIE HEDLUND: Well, I didn't do it myself. We do have other ICANN staff who managed that bit of it, but I did think that you might want to have some place to eat.

So thank you, everyone. We have here Haya Shulman. She's from the Technische Universität Darmstadt. I probably said that really badly. Operational Realities of Running DNSSEC and Cipher-Suite Negotiation Mechanism.

Welcome, Haya, and I will turn things over to you.

HAYA SHULMAN: I have to push the button. Hi. Okay. So I'll talk about the adoption of DNSSEC and a specific aspect to deploying DNSSEC on the name server side, and then briefly discuss Cipher-Suite negotiation, which may solve some of the problems.

Okay. So there's certainly a number of very positive stories about the extent of the adoption of DNSSEC. Some large ISPs, such as Comcast, are validating DNSSEC responses. Many domains got signed – top-level

domains. Generic country codes, more than half top-level domains have signed. Also in the Reverse DNS tree. It seems that DNSSEC is finally taking off, which is good.

But, however, despite the clear tendency showing an increase in adoption of DNSSEC, it is still not widely deployed. Why is that? This is what I will look into today.

I will talk about one of the main reasons in my opinion, and I'm sure many of you will agree with me, that impose an obstacle towards adoption of DNSSEC, and that is complexity and dependencies on cooperation among multiple players for deployment of DNNSEC.

On the resolver side, as you all know, there are many Legacy intermediate devices which may break DNNSEC, but what about the main server side? There was a presentation at [inaudible] I think from 2012 which said – and back then there were much less top-level domains were signed, and so the presentation said that currently today if the zone operators wanted to, more than 80% of the zones could have been signed.

There is a question. Is this really so easy? And what does it take to actually sign the zones? So will talk about the main servers, which is this side of the DNS, for those of you who may not be familiar with that.

I recently did a study when testing the front end, so I recently did a study about how the deployment and how the topology of the main servers are connected, dependencies among different domains and zones.

What I found was that, among the name servers, which is I think a bit less known to the operational community and also definitely to the research community because there's not so many paper published on this topic, the main servers are in fact not a single machine, which seems to be a common believe that name servers are just one machine. But actually, it seems that name servers are composed of a number of machines, and that reminds the resolver side as well.

When you send a query to a name server, if you look here you will the client and the recursive resolver of the ISP, which provides services to the client. That resolver sends requests to a domain, receives a response – referral response – with an IP address, among other things, of the name server.

When you follow that IP address, you actually reach a machine which is a recursive resolver and not a name server. This recursive resolver then either serves a response from its cache or sends a request to the actual name server. I'll tell you in a minute how many of those are out there.

Actually, in fact, often it's not just a single resolver. As you see here, this square contains two machines. This is the name server side. Often, this is not just a single resolver, but actually a chain of resolvers, which forward the requests from one to another.

Here's my drawing showing that here is the first machine which receives the request, from the client – the resolver – forwards the request to another forwarder or a chain of forwarders. Eventually the request reaches the name server which hosts the actual zone file.

Okay, now that's the basic idea. And now how do you find those machines? How many are there out there, and what are the implications of deployment of DNSSEC, or adoption of other mechanisms?

So as you see, our intermediate Legacy machines are common not only the resolver side, but also on the name server side. How do you detect those? I call them, and suggestions for alternative names are of course pretty welcome. I call then RANS, which is Recursive Authoritative Name Server, and that's not just one machine, but the entire set up, which consists of the first resolver, and then the forwarder, and then eventually the name server. So there are those RANSes. And how you detect them?

There are a number of techniques and measurements which can allow it to detect those machines, most of them using side channels. One of them is you send a request for something that is not in the cache of the resolver, so you concatenate a random string and then you sign the query, and then the resolver caches that query. It also checks that all of them use caches. So the resolver caches that query. Then you send another query and here you measure the latency between the client, which is a recursive resolver and the name server and all the machines that belong to the name server.

So you measure the latency, and then you send another request for the same query, which now should be in the cache of the resolver, and you measure the latency, and then if the differences are significant – and actually, in many cases they are because these name servers are often located in another AS and not in the same AS as the first resolver, which

receives the query on behalf of the name server, which is actually the name server because the current domain reports that IP address as the name server.

The differences are – the smallest one that I found – was 30 milliseconds, but actually many more often, and that allows you to identify those infrastructures.

But actually often there are many other easier ways to do that because many of those first resolvers in this configuration, as you can see, the first one – the resolver – they're opened resolvers. Opened recursive resolvers. Namely, they will provide a service to for any domain and not only for the domain for which they are registered. Say they serve food at bar, but if you asked them for your own domain, they will also send requests and serve those requests. So that can also help you find those. So the open ones are all RANSes.

Now, how many are those? Due to time limitations, I will just discuss the open ones, and we'll focus on Alexa 50,000 top domains. But actually I also tested the reverse DNS tree and not only open resolvers, but that's in paper if any of you are interested.

So how common are they? They're about 38% of domains, at least, probably more. I haven't [inaudible] them all yet. In Alexa-50K, a bit more than 6% are open recursive resolvers. I'm talking in the first 100 Alexa domains, there is nothing. There are like standards domains. But after the first 100, you will see many of those. More than 6% are open recursive resolvers. More than 32% are of that complex infrastructure where you see where you have the first resolver, which receives the request, and then forwards the request to the name server or a

forwarder which then will forward to the name server. That's the distribution on the top Alexa domains. These are the percents.

Actually, as you look here, you will see that they're quite common configurations. I'm still looking into why that happens. I have a number of assumptions, though, I'll be happy to share with you.

So what are the implications on DNSSEC, and do they support DNSSEC, and how do you check if they support DNSSEC, actually? I'll focus on the open recursive resolvers and different techniques should be used for those that are not open. You send the request for your own signed zone, which I sent a request to my zone, and I check if I receive responses.

So I send requests with the DO bits in the EDNS record. If I receive the responses, then I know for a fact that the resolver is capable of working with the DNS request enabled with DNSSEC.

Often they're not. From what I saw, there is a difference between being able to parse DNSSEC records and even DNSSEC requests.

If the name server receives a request, then you know that the resolver can process EDNS and the DO bit. Then my name server sends a response with signatures and keys, and if the client doesn't receive that, then the resolver cannot process it.

There is also a question of which of the resolvers in the chain it is because sometimes you have a number of those, as I said, and how do you measure others with such channels?

So what are the challenges on in those measurements? The challenges are to differentiate failures with EDNS versus DNSSEC because some fail even with EDNS. Support of the DO bit in DNS is not equivalent to support in DNSSEC records. Also you need to just differentiate failures with requests versus responses. Identify which of the resolvers in the chain failed – first, second, etc. That's a bit challenging. I use TTL for that and EDNS responses and also timing.

So what's the situation? Clearly as you all know, Legacy devices pose an obstacle to DNSSEC – not only DNSSEC, but any other new mechanism. I found that more than 50% of Alexa-50K that are configured in those configurations that I mentioned. They cannot process DNSSEC. They cannot sign today, even if the zone operator said, "I want to sign today.it'm going to sign my zone." And there are many automated tools available for that, which is great, but they cannot do that because then any client that would send requests to that zone would not be able to receive responses due to those Legacy devices.

Out of those 69%, 39 fail with DNSSEC failure, different error messages from [inaudible] or a server fail. 30% strip DNS records. 18% do not support EDNS at all, and even higher percents are in reverse DNS tree, which is bad because reverse DNS tree is used for security mechanisms, as you know.

So here you can see the graphs for how many – that's just the graph that I showed you earlier – how many of those are there, and how many actually return different types of errors. The red error is the FRMTERROR or SRVFAIL. The black one is "Strip DNSSEC Records." The green one is DNSSEC-Okay/Supports DNSSEC. So as you can see, there

are very few relatively. And that's Alexa-50K. Those are widely-used domains in .com. Most of them are in .com.

So the question is, is it worth the effort, of course? Clearly, it is worth the effort. DNSSEC prevents as attacks, as you know, by man in the middle adversaries, different adversaries, by off-path attackers, as I and others published. It also prevents attacks against vulnerable name servers. DNSSEC also provides evidences. I'm working on something which was recently published, which you could use DNSSEC signatures to actually prove that you were attacked.

For instance, if you don't trust your parent domain and it is capable of signing your records with a different key or it is capable of forging your key, then you would be able to dissect that by sending a query and receiving that, even retrospectively. That's in contrast to other defenses which provide, for instance, encryption.

And of course, DNSSEC would facilitate a number of – many recent proposals for security protocols such as ROVER for routing and DANE, and many others. So we definitely should do it and it is worth the effort.

Now I'm coming to the second part of my presentation, which is Cipher-Suite negotiation for DNSSEC, and that's the work that I'm doing with a Professor Amir Herzberg from Bar-Ilan University.

Okay. So first of all, one of the obstacles towards adoption of DNSSEC is what I mentioned, Legacy devices. Another one is large responses. Those large responses cause interoperability problems with firewalls and how they intimidate devices.

There are proposals to transition to TCP, but not all support TCP, and it also adds more overhead as opposed to [UDP], although actually it does seem to be a good idea to do that.

Everyone is required to support RSA. That's a mandatory support according to RFC, and that also decreases motivation to deploy new ciphers, more like efficient ciphers with smaller signatures and faster validation times, because then you have to anyhow support RSA. Then why bother?

Then you see that DNSSEC signed responses are really huge. I even saw responses with 10K for a DS record in I think under org. That's a lot, right?

So what do you do? In contrast to other security protocols which were defined, such as SSL and [inaudible], DNSSEC doesn't really support Cipher-Suite negotiation. So you send all the keys and signatures that you support – that the zone supports – and our proposal is to deploy Cipher-Suite negotiation without introducing additional RTT to the communication round trip time to the DNS transaction.

One idea is to use the EDNS, and that's the simple extension to the recent RFC, which recommended to signal support of different algorithms to the name servers, and that would allow name server zone operators to decide whether they can actually start deploying new algorithms.

Currently we did deploy that, but we deployed this as separate modules, one on the resolver side and another one the name server side. The resolver sends a request. The Cipher-Suite module adds the

options in the Cipher field in EDNS, and then the resolver receives signatures and keys which correspond to the optimal Cipher supported by the name server and the resolver.

To prevent downgrade attacks, the zone operator should sign all of the ciphers that it supports and the response, and so the client can verify actually that no downgrade attack took place.

But then, as the authors of RFC 6975 just noted, EDNS would not work. It's a transport layer mechanism, and it would not work with forwarders and intermediate devices like proxies and so on, which is right. Also on the resolver side, and as we just saw on the name server side.

So our idea was to add to the design of Cipher-Suite negotiation an application layer. The idea is to concatenate the algorithms to the query. So you can concatenate the algorithms. And I'm just using the numbers here which are associated with each algorithm.

Here is an example, for instance. You can use RSA, RSA/SHA1, RSA/NSEC3/SHA1 and ECDSA as an example, and you concatenate that to the query and you use a delimiter so that the name server, by parsing that query, can identify that the query is requesting DNSSEC Cipher-Suite Negotiation. The name server then returns responses according to the optimal cipher which it selects based on its preferences and the preferences of the client.

But the challenge you encounter to using your DNS is that how can the name server signal to the client which algorithms it actually supports? Because in EDNS as I told you, we use a special record for that in the

EDNS field here, right? So the server would signal with the signature its own preferences.

Now, one option would be the name server can just return its preferences in the query. But that would not work because the queries have to match. So you have to return the same query. The query has to be the same as in the request.

The idea is that's open to proposals and I'll be happy to have your feedback. The idea is to encode the priorities in a dedicated DNSKEY record, which would be signed as the [other] DNSKEY records with the KSK. That would allow the client to check which options, which ciphers, are supported by their zone, and to check that actually no one performed a downgrade attack and it received an optimal cipher.

So that would conclude my talk. Intermediate devices are a pain to adoption of any new mechanism, and of course to cryptographic mechanisms and most certainly also to DNSSEC, as you all witnessed.

This is a common thing in the Internet. But it's on the resolver side, and as we saw, on the name server side as well. They're pretty common [inaudible] and they are likely to persist.

It's difficult to get rid of them. There are different reasons for that. For instance, by using those resolvers in front of the name server, you reduce traffic to the name server. It allows you to distribute your name server to many different ASs by using resolvers which would cache the request, and you reduce traffic, and you also reduce the latency to clients, which send requests and receive records from the cache.

So those are likely to persist, and we need to either upgrade them or design defenses which would work with them. So the conclusion is that more effort is required to speed up adoption of DNSSEC, but it looks positive.

Thank you very much. That concludes everything.

JULIE HEDLUND: Thank you very much, Haya. I see that there is a question from Roy Arends.

ROY ARENDS: Thank you for those two presentations, Ms. Shulman. I have a request on the methodology that you used in your first presentation. If I understand correctly, you do subsequent queries, and if the latency of the second query is less than the latency of the first query, then you assume that the request must have been cached the second time and it responds from cache.

There are additional ways of looking at this as well; for instance, the absence of the authoritative answer bits and response. Or you could look at the decreasing TTL from subsequent responses. Have you included those measurements as well?

HAYA SHULMAN: Not the first one, but I did the second one. The second of looking at the TTL, actually I am looking at that. So resolvers in front of the name server that cached requests, their response from the name server, they do not – at least the ones that I tested – they do not like use fixed TTL.

They reduce the TTL. So by sending a subsequent request, you can see that the TTL is not the same as previous queries. So we did use that.

I was sure the presentation would take me longer, so I did not include those measurements, but not the first one, and thank you or suggesting that.

ROY ARENDS: Okay. And you mentioned in one of the very first slides that you were able to see more than one forwarder. Now, I understand you were able to see at least one forwarder because of the measurements you just did. How do you distinguish between one and more than one?

HAYA SHULMAN: So for all open recursive resolvers, I see that I send the request here to one IP address, and I receive a request on my name server from another IP address. There is often those two machines – those two IP addresses – are located in different ASs.

To identify more than that, it is challenging and it's still work in progress, but what I do is I use the TTL with queries and noticed that actually many of those in the chain are not open recursive. So even if the first one is open recursive, it doesn't necessary imply that the remaining ones in the chain will be open recursive, and most of them don't actually respond to queries.

So it is challenging. I'll be happy to hear suggestions, such as Roy suggested – thank you – with the authoritative bit, and it's still work in

progress. So the main technique that I'm using as I said is timing and the TTL.

ROY ARENDS: Thank you. This was not criticism, just curiosity. I'm very much looking forward to papers on the subject. Thank you.

HAYA SHULMAN: I'd be happy to send.

JULIE HEDLUND: Additional questions, please?

ALEXANDER MAYRHOFER: Thank you for the presentation. When I saw your measurement methodology, I was also thinking that there are certain name servers, especially those with a database back end, that might expose similar behavior. For example, PowerDNS, as far as I know.

The first query is pretty slow because it's fresh from the database, and the subsequent queries are stored in memory. So you might look at other methodologies to filter out those.

HAYA SHULMAN: That's a correct comment, and that's why I'm not using just one query response. I'm using an average time, sending l many – say 10 or 15 – queries, and for, say, a number of random subdomains, and then I

measure responses, and then I take the average. So I did that and I did notice what you were saying. Thanks.

ROBERT MARTIN-LEGENE: Question. Robert Martin-Legene from Packet Clearing House. In these tests, did you go out to the authoritative name servers of these Alexa domains and do a query for the domain that they are authoritative for, or just query for our own domain.

HAYA SHULMAN: The open ones?

ROBERT MARTIN-LEGENE: Well, basically the ones that you say they have a back end chain.

HAYA SHULMAN: Both. If it's an open recursive, then I send a request to mine, also my name server, because it allows me to learn much more information about how they work. I'm also measuring security of those entities, so by using my own name server, I can see a lot of interesting things.

But for those that are not open, I'm sending requests only to the name servers for which they are registered as authoritative.

ROBERT MARTIN-LEGENE: And that request is the one that you say takes longer the first time?

HAYA SHULMAN: Yeah. You can see it here. The first time takes [inaudible] milliseconds, for instance, and the second one in number, like a set of requests, as I said. When they are not in the cache, it takes a certain time, and then when you send requests for something that you already requested, it takes a completely different time. You could see the differences even in, say, 100 milliseconds and more because they're often located in different ASs because they're like in chains sometimes.

ROBERT MARTINE-LEGENE: Have you tried to make a comparison about if there are certain ASs that do it in a specific way, and if maybe that represents, let's say, 5% of the Alexa domain that you were testing? Because like it could be that maybe it would be Cloudflare or something that you went into, and then you're basically making a big study about two or three content providers, maybe.

HAYA SHULMAN: That's an excellent comment. Yeah, I did think about that and I tested them. So you're right. Actually, I found that many of those were located in the United States and China. Those are the most ones with the highest percent. But they're not the same thing – not like one, a couple of entities which run those.

This is a valid point. Initially I was thinking maybe those are the CDMs that like everyone is using. It seems to be a common practice. There are a number of such organizations, but it's a very large study. It would take me more time to characterize them. I'm in the process, but I did see that it's not like – I agree with your comment, but it's not that this

requires explicit checks, but they did already see that it's not just a couple of entities which run those and then everyone just takes services from them. If it were the case, it would have been much easier to fix, of course.

ROBERT MARTIN-LEGENE: The only thing that really makes me consider something weird is going on here is that the behavior that you are suggesting is that there's kind of a broken name server that has been set up like eight years ago and is still there.

But on the other hand, you suggested there's an elaborate chain of back end processing going on, and I don't see how those really work together, so if you could continue studying on that, it would be – maybe you could contact some of the biggest ASs and see what's going on.

HAYA SHULMAN: Yeah. So you're right. I did receive [comments] from the operations community that this is a non-problem where you have host which is configured as a name server and as an [inaudible] resolver. But as you can see, these measurements I actually have all the [captchas] for those.

And if you're interested, I could give you IP addresses which you could test yourself. They show that this is not a single machine which is just a misconfigured DNS software which runs also as a recursive resolver and as a name server.

But this a resolver which then sends requests to the name server, and sometimes it's a number of resolvers in a chain, similar to the client

side, where you also have resolvers and sometimes a chain of forwarders.

So it does require further tests. I'm working on that. But the point is there's so much to study here, both security-wise and measurements-wise and implications of difference defenses. It does shed light, for instance, on deploying defenses on the name server side against distributed denial of service attacks, not only DNSSEC, and it does take time.

So I'll be happy, by the way, to collaborate with anyone who is interested. I'm working on the security aspect and also attacks and also on measurements. If any one of you is interested is looking at the data that I have or running some [inaudible] work together, please let me know.

JULIE HEDLUND: Actually, we have a question in the chat room, and then I think we do need to cut things off. Lunch is behind us but if you might be available for questions during lunch, that'd be great, too.

But in the chat room – and actually it was a separate chat that was sent to me – it says, "Does your negotiation model follow the prefix of –" well, actually, there are two points. Sorry, I missed the first one. Hold on.

It says, "How do you take care of RTT in your second presentation, and does your negotiation model follow the prefix of suffix [algo]?"

HAYA SHULMAN: What? Sorry?

JULIE HEDLUND: Yeah, I don't know either, but you can see it here.

HAYA SHULMAN: So how do you take care of the RTT? [inaudible] so my idea is not to – here as you can see, the client, which is the resolver, and there is a name server and that's two boxes that we did for our implementation defined, our goal is to actually implement those in the real resolver software and name servers.

The RTT, the point is – you send a request, you receive a response, and the question is whether you need to send another request to receive another response, and that's what I meant by RTT. So this doesn't add another RTT.

But IANA introduces problems, as Steve Crocker pointed out, with forwarders, and did. So we then encode this query. I'll be happy to hear feedback from you for that. This is very interesting to me what you would say and whether you would suggest to use a different idea to encode priorities of the server for instance in a DNSKEY record or in another record. But this mechanism also should not add an additional RTT, and the idea is to send a request and to receive a response within the same query.

UNIDENTIFIED MALE: Julie, my comment was just going to be thank you, Haya, for bringing this research here. I think from the Program Committee, I think we were

intrigued to see what you're bringing here. I would also encourage other people, both here and listening remotely, to think about what other research can be brought here to share with a larger community because these kinds of issues around metrics and other pieces are definitely of interest to help us understand the larger picture of what's happening out there with DNSSEC. So thank you for this work.

HAYA SHULMAN: Thank you very much. Thank you for inviting me to come and give this talk. It's my pleasure.

JULIE HEDLUND: Thank you. There's lunch at the back. I would suggest perhaps that we could try to come back maybe 10 minutes before –20 minutes after 1:00 – just because I think we might need a little bit more time for our afternoon panel because we do have some very interesting demos, and we do want to make sure we have plenty of time for them. But since we're in the room, you can't wander far.

UNIDENTIFIED MALE: We should also mention there is not a Great DNS Quiz this time. I know, I'm sorry. Everybody points at Roy. Well, we need some help generating some new questions and new things, so if anybody would be interesting in helping create the Great DNS Quiz for LA for ICANN 51, I would love to talk to you. Come find me. Thanks.

If people could start gathering around the tables, that would be great. Because of some of the demos, we'd like to try to get started a little bit earlier. Thanks.

If people could please find your seats, we would like to get started a little bit early just because we have a lot of different things going on. So if you could please come to your seats, that would be greatly appreciated. Thank you.

Wow, look at that silence.

DAN YORK: All right. Thank you, everybody, for meeting here through lunch. We still got a packed room here for the folks who are remote. I'm Dan York and I'll be moderating this second half of the session, where we have a group of different presenters who are going to be talking about some of the actual use cases and giving us some demos of products that are using DANE and DNSSEC in different ways.

Each presenter has about 10-15 minutes apiece, depending on what they're doing on there. We're going to start with Guido Witmond. He's going to be giving us kind of a use case of DNSSEC and DANE in phishing protections in the SMTP space.

Then we've got Willem from NLNet Labs, who's going to talk for a bit about the getDNS API, which I referenced earlier and is a new API.

I guess I should ask: how many people here have looked at the getDNS API? Paul Hoffman doesn't count to my right since he helped write it – or the spec for it. Okay, good. So we'll do that.

Willem's also going to talk about some of the work they did measuring DNSSEC validation using RIPE ATLAS, which is a cool project on that regard.

Paul to my right is going to talk about DNSHarness, which is a good test framework that you'll hear about.

Then we got Iain, who's going to do a live demo, which we'll see how that goes here, with a DANE-enhanced version of OTR messaging, which I'm very much looking forward to. This is kind of a cool thing in that regard.

Then we're going to round out with Joost from SURFnet doing a demo of some work they've done.

So that's what we've got today. We're going to get started. We are going to try to keep it somewhat tight in terms of time. Please feel free to ask questions. I will take questions in between because they are different discrete demos that we'd like to do. We may also use that Q&A as a period when we're setting up for the next one. But I will try to keep it somewhat tight in terms of that.

We do anticipate having a bit of time at the end where we'll be able to talk a little bit more, though. So without further ado, I would like to have Guido begin.

GUIDO WITMOND: Well then, thank you very much. My name is Guido Witmond. I'm a computer programmer by profession and DNSSEC hobbyist in my free time. Yeah, I like DNSSEC. I like DANE, but yeah, those things on itself

you [need to] use it for something, and stumbled upon a way how to use it for phishing protection.

Phishing is targeting humans and right now the computers don't have anything to protect you, so that's the goal for my presentation – to show how the computer can protect the user.

First of all, we need HTTPS and we need CA. But there's a lot of CAs and everyone is just like the other. If you have a certificate from a site, you can get a green bar and that makes the victim believe that he's connected to the right site. So on its own, it's not enough.

So about CAs. Any of you please raise your hands if you know the CA of your bank. Well, [inaudible]. It's the normal quota. You get very little people that verify these things, so we need something to verify for it.

That's what DNSSEC and DANE does. It verifies if you have to correct the certificate for your website. The person who just raised his hand is clearly not using DANE, because otherwise the browser would have done [inaudible]. Sorry.

This opens the door for new possibilities. This is just a slide on how DANE works. It's not for this audience. In short, DANE specifies what the user can expect, and the browser does the validation to see if it's correct according to the expectations. When these expectations are met, you are on the right site, and if these expectations are unmet, then the browser shouldn't show anything.

So we've solved the CA problem. Now you can choose your CA based on what you really want to do. I think CAs should be really happy with DANE because they can differentiate from each other.

If you don't use the TLAA verification or validation, then you're out of luck. You get [inaudible] in the Netherlands.

Scammers and phishers can set up a bank, just like a normal bank. They can protect it. You don't have to validate anything, but it's the wrong site. If they get your user name and password, you're out of your money. That's the problem: users shouldn't manage their own passwords. You should let the computer do that.

So we're going to do that. We create a password manager. It manages all the accounts. It does the log in. It does the log out. It has buttons for these operations on your browser frame. It doesn't let users write passwords into sites. So the password manager does all the account management and the user only specifies what to do.

A password manager does one other thing more. It does DNSSEC validation because it needs to know if the site you're connecting to is the correct site, because if it's the wrong site, it will provide the password to the scammers. So you would take it out of the operating system into the application level.

Now this is what makes phishing difficult. Scammers can copy the site and then can make it look perfect, but it can't fake the domain name because the registrar won't sign the same name twice, and DANE prevents another registrar from signing that same domain name. So the scammers really have to use another domain name and see if they're lucky.

A really scared user – someone who is really scared that the fake site is the real site and something really bad is about to happen – probably

wants to get the password and give it to the bank; in fact, give it to the scammers. Well, that shouldn't happen, even if the user does something stupid that he shouldn't give his credentials to anyone else. So the password manager should not reveal the passwords to the user, and the password manager and bank best would use zero-knowledge proof, so even if the user does something stupid, the scammers don't learn anything that they can abuse.

A few more things. People can have multiple devices. Well, for that we have browser synchronizing. Yeah, your browser or your user agent is storing all your passwords, so that needs to be protected. For that, you might use a password.

On the count of malware, yeah, malware is still a problem, but it's a problem for everyone. This problem is not included. So if we can stop malware from getting your computer, we're already way off better.

I dislike passwords with passion, so I really avoid them whenever I can. Instead of a password manager, I use client certificates. Client certificates make my life easier to build it because web servers already know how to deal with these things. It really makes it easier to program.

So on one hand, I have a user agent. It requests client certificates, and on the server side, I have a certificate signer that signs the client certificate. Each site, each bank, they sign their own client certificates, and they only accept their own and not those of others. That's an important part. Otherwise they could accept their own client certificates for someone else.

I have a little demo network. The left is the browser. Next to the browser sits the user agent. I call it Ecca-proxy. It manages the accounts. On the right hand side, I have two banks: the phish-free bank (that's the good guys) and the scam-full bank (that's the bad guys) and they publish their address records and DANE records in DNSSEC.

Here is the browser to the good guys. On top of the address bar, very little in gray, it says Phish Free Bank before my domain name. So this is where it starts. I start by opening an account and the bank says, "Well, we use this client certificate, so you have to use the proxy, and if you've installed it, you can click here."

Now, this is a page from the authenticating proxy from the user manager. It says here on top that it doesn't have any identities online. True. I haven't been to the bank. I just registered. I choose my own name and I register.

Here on top is still the authentication manager, and on the bottom is the output of the website. It says that I'm logged in at this site with this account name. The bank opened my account at this name and it gave me an account and even 25 monetary units. There's my transactions.

So I log out. I come back to the site again later, and it says, "Okay, you already have an account." I click on it and I'm logged in there's my auto-transactions.

So one fatal day, I get an e-mail and I get scared by it. I click on it, and instead of going to my phish-free bank, which is the good guys, it takes me to the bad guys. That's the sign of the bad guys – [you can't] spot

the differences. Well, I'm phished. I don't see any differences, so I think this is my good site.

So I try to show the transactions, and the account manager says, "Well, you need to authenticate," but, oh, it says none. Well, something may have gone wrong. Let's type my name in again and register again. Okay, I'm in. Where am I logged in now? It's the bad guys because my user agent already knows that it has an account for phish-free bank, which is DANE-validated, so they can't abuse that name. So the scammers have another bank. The computer prevented me from logging in with my credentials at my bank and instead the scammers still have nothing.

So DNSSEC is good. It makes DANE possible, and with that, a different way of handling account handling on sites, you can reduce phishing. With that, I end this talk.

DAN YORK: Thank you very much, Guido. Anyone have questions for Guido about the demo he showed here with his account manager?

Okay, well thank you very much, and we'll have Guido sit here for the quick Q&A at the end of that. All right, let's go on and bring Willem. Do you want to switch? Okay.

Willem is going to do two presentations. The first is around the getDNS API itself, and then the second one will be about measurements. This is the getDNS API.

WILLEM TOOROP: Yes. The getDNS API is a DNS API specification for resolving names, and what's special about it is that it is designed by application developers and also designed for application developers in a process led by Paul Hoffman.

Verisign Labs collaborated with us from NLNet Labs to create an implementation for it, and more people. No Mountain Software was also participating from the beginning, and now Sinodun is also joining in.

DANE applications need to do more encrypting. They have to set up encrypted channels. But to be able to do that, you have to authenticate the public key of the other site to protect against man in the middle attacks.

The current solution, PKIX, is not really convenient. There's a certificate authority repository that either comes with the application or the OS, but it needs a lot of administration. You have to [inaudible] list and all those sort of things.

But most importantly, every certificate authority is authorized to authenticate any name. So you could say "Well, I trust my certificate authority. They are trustworthy." But that's not enough. You have to trust all certificate authorities to authenticate your certificate.

It's a weakest link problem, and DANE provides a solution for that by putting a finger in the DNS pointing to the specific certificate authority authorized to authenticate your public key.

This picture – the cartoon – is made by Olaf. It illustrates that. Or you put the certificates right in order [inaudible] of the certificate in your

DNS. You actually only have to trust the top-level domain that you pick yourself and the roots.

This is all very great, but applications cannot make use of that easily because the [inaudible] function to give you DNS answers, get info, give you only addresses, and not DNS resource records that are specific to DANE, TLSA, or security fingerprint.

Even more, you need authenticated answers for that, and maybe the resolving in the network is validating if you're lucky, but you do not get a [dbit] from get other info.

Also there's the last-mile problem. Maybe the network recursor is validating, but the network itself is not very secure. It could be [inaudible] spoofed by a malicious resolver, or otherwise you could be sitting in a pub with Wi-Fi. It doesn't have to be a controlled local network.

The best solution to this is to put the resolver right at the application to link resolver [inaudible] application and bypass the resolver completely. But DNS works so well because it is distributed in many different ways. There are authoritative sides of distributions, but also the resolver side is distributed. In this way, we would bypass all the caching that is in DNS and that's why our library also has DNSSEC integration as a step. It just asks the local network resolver for everything needed to do the DNSSEC to validate the DNSSEC chain itself and deliver the authenticated answers to the application.

The nice thing about it is that your local network resolver doesn't even have to do DNSSEC validation. If you have a system administrator that's

not aware of DNSSEC or not interested or he doesn't trust it, then the application that wants to set up encrypted channels using DANE, authenticated with DANE, can just do it from the application with a DNSSEC-aware resolver. In the presentation after this, I'm going to dive deeper in how common DNSSEC-aware resolvers are.

This is from the specification written by Paul Hoffman. This is the motivation for the application developers and why they needed a different API, a different library. They wanted to have a natural follow-on to get our info that was not written by DNS people and didn't do everything that is possible with DNS and getting to all of the itty-bitty details of DNS, but provides you the things that are needed for applications and nothing more.

First publication was in April 2013, Creative Commons. We implemented it and released it first in February, and the 0.1.3 release is about to happen. We have no JS or Javascript [inaudible] which we're very excited about. It is a license that allows any applications to use it.

This is why I think that the getDNS resolver library is a good thing for your application because it combines parts that were before only available in different libraries, and it does that by linking to those other libraries.

It also gives you a very generic data structure consisting of lists, data and integers. If I ask this question, "How does it look? What can I do with it?" for trial-and-error-style programming, it's quite popular and also really good to modern scripting languages.

You're going to have a look at how that works now. This is an example with the python bindings. This is how to query for a TLSA record for getDNSAPI.net domain with the getDNSAPI implementations with [inaudible].

In a full recursion mode, this how to do it in stub mode. You see that you have to create a context-to-context that contains all the caching data of the full resolver, and also the modus operandi. You can it in stub mode also. It contains some other configuration parameters for the resolver, like number of seconds to timeout and that sort of thing.

But as I told you, you need to have a DNSSEC-aware resolver in your network to be able to do this. So the best way to find out if your local network resolver is DNSSEC aware or not is to ask for the root key and see if you get it signed, and it's probably signed. If not, create another context bypassed the local resolver and use the full recursive [inaudible].

At the bottom, I hint at what the response dictionary will look like, and I'm going to share with you here. It looks like a piece of JASON data. It contains all the full replies and everything you might – this is an answer to a request for address data for getDNSAPI.net. You get the full replies of the raw packets. There is a convenient key to get to the addresses quite quickly. You have the status and then there's the replies tree, which contains the actual DNS packets that were returned. It might be multiple, because as you saw, it returned both the IPv4 and IPv6 address.

I've put on a CGI script on our website, and you can perform a query yourself with the API and have a look at the response dict to see what

you can get out of it and also try out specific extensions, which are the checkboxes you see.

This is what we now support. We make sure it runs on these platforms. We're working on Windows and Android. Packages are in the make. There was a silly mistake in the 0.1.2 release [before] which we have recently fixed today. The 0.1.3 release will come out and then we get at least the Debian package.

We are already in FreeBSD ports, by the way, and there's also a homebrew package for Apple. You can get it here. It has a few dependencies: libunbound to do the actual resolving of DNS for getting to the elements of the DNS.

Also, as requested by the application developers, libuv is quite good in performing asynchronous lookups, and it links against these popular event libraries. It's modular. You can choose.
If you like libuv, you choose that measure and do the asynchronous lookup if you're in the library.

Okay. We held a Hack Battle in Amsterdam in April to get feedback from actual application developers to see how well they did with our library. There were four teams. The first one created a plugin for Thunderbird that verifies the credentials of the DKIM records associated with the e-mail to be more certain that DKIM's statement is actually secure or does make sense. They actually won the prize in this Hack Battle, as well.

Then we had DANE Doctor. They did a DANE client library for the popular part and event framework – Twisted, an asynchronous event framework.

We're going to have a live demonstration on this next, so I'm skipping this quickly. And then we have the DNSSEC name and shame, which by the way is not only DNSSEC signed – this name – but also the website has a DANE record.

Also, if you're checking domain name that is good, who gives you the thumbs up now? Yeah. Can I get a T-shirt as well?

So yeah, this is the first getDNS presentation here. All the details to get to the repository our mailing list the specification. The movies for the Hack Battle. Questions later, I think?

DAN YORK: Are there any questions about Willem for this? Because he's going to move on to something very different. What do people think about the getDNS API? A couple people have used it. People interested in using it more out of this? Yes, good stuff. Okay.

JULIE HEDLUND: We have a question from the chat room. It's from Mark Lampo. "In order to validate the public part of the KSK, the root must be known to the package. How is this configured, and particularly how will it cope with a root KSK rollover?"

WILLEM TOOROP: Well, we don't actually package the root key in our distribution because Verisign felt a little uncomfortable with that. But it's used as unbound under the root. There's this tool – unbound anchor it's called – which

does everything that's needed to get the root key secured in a place where they getDNS live we can make use of it.

Also because unbound is under the root, it also take into account root key rollovers. It follows RC5111 I think it is for safe [rollover] of the root key.

The package maintainers, like the Debian and for FreeBSD [ports], they are the ones that have to make sure that unbound anchor is called after their getDNS package is installed, equipped library with root encryption.

DAN YORK: Geoff Huston?

GEOFF HUSTON: Failure's always more interesting than success.

WILLEM TOOROP: Yes.

GEOFF HUSTON: So you go into this mode where you're trusting this recursive resolver and it says, "Find question. Server fail." What do you do?

WILLEM TOOROP: Yes. Very good point. In the python code I illustrated – let's go back there – it has the "if the status is getting a response data [that's] good – " before you see the extension, DNSSEC return only secure, and then at the end you see if everything is good, then make use of it.

So if you get a bogus answer, if it's not good –

GEOFF HUSTON: Serve fail, which is not an answer at all. It's just the DNSSEC validation failed answer, but it's encoded as rubbish, really.

WILLEM TOOROP: Then you may not connect with that server.

GEOFF HUSTON: Even though there might be other NSs that could give you the right answer?

WILLEM TOOROP: If other name servers that do not give a serve fail, I think also in unbound under the root tries everything it can to get an answer.

GEOFF HUSTON: Ah, so you're relying on the internal implementation of unbound to go, and so whatever unbound does, wherever they are, is kind of how you've been abstracting that.

WILLEM TOOROP: Yes.

GEOFF HUSTON: Okay, thank you.

WILLEM TOOROP: Yes, but what is good to notice though that if you get a failure, you may not connect with the remote host [inaudible].

DAN YORK: Behind me, are there any questions behind me? Okay. With that, I'd like to go on to the next presentation about measurements using the ATLAS network. We will have time at the end to have a little bit more question and answer, so think of questions you may have that you'd like to ask any of the folks who have been presenting up here.

Remote participants as well, we're getting set up for the next presentation, and I notice some of you are asking questions in the chat room, and we will, as Julie has done, try to relay those as we can. If we have time at the end, there have been a couple comments too in the chat that we will try to relay as well.

And now, back to you, Willem.

WILLEM TOOROP: Yes. On the location where we're at – close to the University of Amsterdam – they have this system in network engineering master, and they do one month research projects, and we quite often have a student doing a project with us. One of those projects is measuring DNSSEC-validating resolvers in RIPE ATLAS, which is performed by Nicolas. This is his picture. As the report is not yet done, there are some corrections to be made in the report, but he spent one month using RIPE ATLAS measuring the resolvers that do validation.

He gave this presentation. It's right, but I adapted it a little bit to put more focus on DNS-aware resolvers because that's interesting in perspective that I just gave with the getDNS presentation to do DNSSEC iteration in stub mode.

So ATLAS is a network of measuring probes. They are handed out by RIPE at conferences and at other locations and there are more or less 6000 of them worldwide now, mostly in Europe and quite a few in the United States.

It can be used to do all sorts of network measurements. There's probes typically in just the clients' network or in the people that have a probe, they get them for free. They put them in their own network. The probe gets a resolver with IP address and resolver by DHCP or otherwise, and thus everything looks into the Internet the same way their computers would – or at least that's the idea.

So it gives you, compared to other measurements that have been done before by Geoff Huston, for example, the difference is that you have a look from the inside into the exact details of the packets. You see exactly what is returned when a certain question is asked.

This is really interesting when we want to investigate how many DNSSEC-aware resolvers are present in RIPE ATLAS and what do they look like? Can they actually be used to do stub-level validation?

In the setup, also the queries were captured on the other side. There was a name server controlled by Nicolas, and he sent out queries to the probes to be executed, and also captured what he saw on the name server side and compared that.

What you see on one side at the probe and what you see at the authoritative side is a completely different thing. Anything happens in the middle. You have Knot of course, but also forwarders to forwarders. It's very hard to associate those two things, but it's possible with RIPE ATLAS to prepend the probe ID to the query, and also the measurement ID. So you can tell exactly at the authoritative side which measurements are these and which probe caused the query.

Oh, there's also that ATLAS is certainly not the Internet. Here you have the ATLAS top ten probes. United States is 1, Germany is 2. This is very different from the actual Internet. We have some ideas on how to make the outcome more realistic, but we have not done that yet.

This is the process that we did. We listed our probes, started to capture on a name server, launched measurements, waited for the results, stopped capture, and analyzed.

The first thing Nicolas did was just ask for TXT record and see which resolver set a DO bit, and that's 88%. 67% of the probes also received RR signatures.

With this measurement, he also could make a resolver distribution. Many probes use a few resolvers and fewer and fewer probes use less resolvers. This is this distribution.

This is I think the more interesting table. Because we prepended the probes' ID and also the measurement ID, at the authoritative side, we served a wild card. So we asked for a wild card. What we noticed was those zones that are answered here – the bad label – were all bogus. There's a secure zone on the top and the rest of them is bogus. So the

probes that do get DNSSEC protection in some form should not get an answer with the bogus zones.

You see here in the lower half of the slide that 26% on average do not get an answer with bogus zones. So for RIPE ATLAS, 26% is protected by DNSSEC.

But what's also interesting is who does get the signatures, because if you also get signatures in your answer, then you do not need to be protected. The stub can do the validation itself.

On the top half of the slide, you see RRSIGs. There are two types of RRSIG columns. There's RRSIG plus the NSEC, and the RRSIGs only. So if you add them, it makes 36% or so of the probes, besides the ones that serve you with servfail, and an additional 36% give you the signature, so the stub can itself decide that an answer is bogus or not.

Except in our case, by asking for a wild card, it's only 90%. In 90% of the cases, we can actually validate that because in a wild card answer, you also need to prove that the actual name does not exist, and 90% of the resolvers we asked do not provide the NSEC with the signature. So you'll end up with a bogus answer, even though it's not.

But we thought this was quite an interesting phenomenon, so to be able to tell if a name exists from a validating stub, 65% when you have a zone or ask for a name that's not a wild card, but only 46% when it is a wild card answer to a wild card.

We also had to look at DS support. If we ask for DS, do we get the answer? Because DS is out of zone data for normal DNS operation. The parent is authoritative and not the client.

So by simply asking for DS, we get 95% answers. But that's not really fair because if you do normal DNS operation and you come by a parent that has the DS, it might give you the answer. But if the resolver would have the DNS records of the client zone cached, then it might not be able to give you the DS answer because it will actually query the client zones name servers.

So we did a second measurement by first creating a completely new zone with a completely new name, then first query something other than DS to cache the NS records and then ask for DS, and we still get 93% of those queries answered.

This is interesting if you want to do everything, all the DNSSEC validation from a stub perspective, so not only if you get RRSIGs but also ask for the RRSIGs in [inaudible]. We are not actually considering it, but it was still interesting to see if it is possible.

So we think that because we also saw with the simple TXT queries sent a DO bits [when] they get a delegation. Because they set a DO bit, they also get the S in the authority section, and most resolvers just cache everything in the authority section – or at least some. That would explain we think the high percentage of answers here.

We also had a quick look into a non-existent answer. This is quite consistent with what we saw before. We have the 26% that actually validates, and the 36% that is DNSSEC aware. So that adds up to 65% that delivers RRSIGs.

What was also interesting is that 25% of the answers were spoofed. Those were OpenDNS for example. They hijack your request and

redirect you to a website saying, “Oh, so you’re interested in this and that? Well, have a look at our advertisements."

Also if had a good few, as my earlier results, we see consistently that 5% gives forum error. Those 5% do not give that if there is not EDNS records in the request. Those are resolvers that cannot handle EDNS error.

So summarizing, 26-28% of RIPE ATLAS, in a rather biased environment, does validation. Also, it’s about the same amount that gives protection. 65% gives you enough information to do DNSSEC validation at the stub site, except when you’re doing wild cards. Then you’re down to 46%. That’s it.

DAN YORK: Thank you, Willem, for this information. I appreciate it. I always wondered what was the penetration within the ATLAS network. I run one in my own basement in New Hampshire in the United States, and I think that what we see there in terms of that does show it would tend to be the geekier side of folks who will run the probes in there, just seeing the 26-28% validation is certainly a bit higher than what Geoff’s research has found in his Flash-based work that he was doing.

Geoff’s shaking loudly his head. But I have a question or two questions for you. Do you have any intent to do this research on a more ongoing basis?

WILLEM TOOROP: Yes.

DAN YORK: Okay. One of the things – and I'm looking at you and also looking at Geoff – one of the things I would personally love as an advocate for DNSSEC and trying to do this is I want the DNSSEC validation equivalent of Google's IPv6 chart that shows a nice little track that shows how IPv6 penetration is going along. I want that kind of chart from somebody that can show the ongoing deployment of validation that we can watch over time that we can use as a reference and a trend line.

So I guess my appeal would be to anybody running the right probes – I'm realizing it's biased, but still – or Geoff – I'd love to see a chart like that that we can start to use within industry to show where are we with DNSSEC validation.

Geoff's chomping at the bit, so I better shut up and let him go.

GEOFF HUSTON: I have an answer for you and question over there. The answer for you is [inaudible] .Rand.APNIC.net/cgibin/worldmap.

DAN YORK: You're just like Anne-Marie, okay? We'll get that out there.

GEOFF HUSTON: We'll get that out there, but there's one out there. I've done it.

DAN YORK: All right. Is it on an ongoing basis?

GEOFF HUSTON: Every day.

DAN YORK: Every day?

GEOFF HUSTON: Every day.

DAN YORK: Awesome.

GEOFF HUSTON: The question over here is actually a really big question because I must admit I started down trying to answer the same question: how many resolvers are DNSSEC-aware? Because of the fact that there's no snail trail of a query as it moves through forwarders, I couldn't tell the difference between a validating resolver or a non-validating resolver that was having queries forwarded to it from a validating resolver.

You get this real problem that, quite frankly, when you start looking at resolver behaviors, they behave as if they both are resolving and not depending on what's behind them.

I gave up because I couldn't answer the question, which is why I went back to the same thing that Google is saying about v6. How many users won't resolve a name because the DNSSEC validation is broken? Which is ultimately the real test of DNSSEC. If the sig's bad, it shouldn't resolve.

So you're kind of asking how many users (A) go through the dance of doing validation and (B) when they get back the serve/fail indication, do not go any further.

I find this work very difficult to reconcile because (A) I don't know about resolvers because they hide behind other resolvers and (B) there are some resolvers, and there are few in Comcast, that handle, ooh, a couple of hundred thousand users; and some, like mine, handle me. So the fact that I resolve doesn't make an ounce of difference in the world, but the one that Comcast does is actually quite influential.

DAN YORK: Yes. So that was my point.

WILLEM TOOROP: Yeah. So what we do – this also should have been described in the metrics as well – but the ATLAS probes give you an answer for every resolver they get configured with DHCP. So what we have done is – maybe it's good, maybe it's not, or what Nicolas has done – is take the first of the question, if you get the serve/fail, fall over to the second resolver of the probe. If that fails, fall over until you get an answer. Or if none answers, then it's a failure; it's bogus. With that process, 26% was protected against bad data.

DAN YORK: Okay. I think there's some more questions we can have around that later. Geoff, I did find your site. You're right. I want the swoopy trend line, though.

GEOFF HUSTON: Yeah, you want me to muck with the data.

DAN YORK: No, I want a trend line because you have a map and it shows everything all red.

GEOFF HUSTON: No, just go down to the first country called XA, which is the world.

DAN YORK: Oh, XA is the world.

GEOFF HUSTON: [inaudible] like that and that's a little graph, right?

DAN YORK: Oh, okay.

GEOFF HUSTON: A trend line that goes up.

DAN YORK: Oh, all right. I like that. Thank you.

GEOFF HUSTON: You'll also notice by the way that it plots the number of people in the world who use Google's Public DNS. That's the other line. That's the line that's higher than the number of people doing DNSSEC validation.

DAN YORK: All right.

GEOFF HUSTON: There's a story there, too.

DAN YORK: Yes, there's lots of those. But speaking of stories, I want to bring up Paul Hoffman to talk about DNSHarness for DNSSEC. Take it away, Paul.

PAUL HOFFMAN: Hi, I'm Paul Hoffman. So what we've been hearing is some research that people have done, and what I'm going to be presenting on is how you can do your own research, a certain kind of research, using an open source test package called DNSHarness that the development of it was supported by Verisign Labs.

I'm going to do a real quick overview of it. I'm not going to demo it because running it is really boring, as a lot of research often is. But I'm going to show you how you can get it, how you can get the package if you are a researcher yourself – am I doing it or are you doing it?

JULIE HEDLUND: You can do that one and I'll do this one.

PAUL HOFFMAN: Very good. Okay. So let me just do a very brief introduction to DNSHarness. It is a system for testing queries where you can take one query or a set of queries and sending them to lots of different either authoritative or recursive servers. So what this is testing, what it allows you to test, is servers. You can set up any kinds of queries you want. You can put things in between you and the server, but the idea is, if you want to see what happens when I send this kind of query to servers, and all sorts of servers – every version of BIND, NSD, and things like that – this is the harness that you want.

One of the advantages of this harness is all of this open source software comes pre-built in it. You've got like 400 different images. You've got every buildable version of BIND, which isn't all versions of BIND, including many from version 8.

So if you're a DNS researcher, this is for you. This is probably not for you if you just have a simply question. But if you are researcher, this might be something you're interested in. It runs on a LINUX box. It takes its own box, but that's all you need. You don't need a whole set of box or anything. Everything is done in virtual machines there.

So it's useful for any kinds of research you might want to do based on requests, including things like DNSSEC, new RR types – some people have said, "Oh, we can't deploy this new RR type because it won't work in some environments." This is a way for you to determine that.

The other thing that comes out of using this test harness is, once you publish your results, you can also publish exactly the way that you got

your results. So other researchers can take your research methodology, change it to their desires, and go forwards. So this is actually fairly good for "I think this is true," and then some other researcher can say, "I did what you did, but I did it a little bit different and I got a different result," just as we were hearing with Geoff and Willem.

So it's basically just a python script that's running in a LINUX box. That box is running VirtualBox, and in VirtualBox, all the open source DNSs live in one giant VM. The advantage of that is, when you start sending off queries, you essentially say, "Start up this version. Send the query. Shut it down. Start it up. [Shut it down]."

You can also put other things on that box. You aren't limited to just what it comes with, so if you have for example Microsoft Windows server or things like that.

You can also send your queries off-box. You can use that box as a NAT, as a broken firewall, as a good firewall – any of that. You add your own your projects. You run the projects on the box and I'll show you an example of how to put together a simple project.

Again, this is useful if you're interested in all of the open source software out there, but also if you want to test real world things outside.

Since we're talking about DNSSEC here, some of the things you might want to test as a researcher, such as does a certain recursive server properly report the status if an upstream is broken, if there's something broken it between? You can even put something breaking in between. Hopefully all of your answers will come back saying, "Yes, there's

something broken, but of course, we know not all of the DNSSEC resolver libraries pay attention to the errors as much as they should. So this is a way to find that out.

You might want to check if an authoritative server sends the correct records if the request is broken in certain ways. For example, you might send – and this is a common thing that we're finding – if somebody is trying to show how to get positive DNSSEC answers from bad queries, this is a way to do that. And you can use this for fuzz testing and things like that.

How do the various recursive work if you've just done a key rollover? There's lots and lots of stuff you can do with this. So these are the kinds of things you can do, again, if you're controlling the query and want to see how the servers respond.

So a project directory, which is you build your own project. This is just a little JASON object. I'll show you an example of one. Then there's two programs. One that runs on the LINUX host, which is the thing that's sending the queries. It says, "Start up the server. Do that." Then the one that actually sets up and tears down. Since there's examples of it. But if you add your own servers to this, you can say, "Okay, to start up Windows server for example, I actually need to boot the box and wait for a while to do that." All of that's easily configurable.

So this is an example of a project description file. It's just a little JASON object that you can ignore the first two, and then the targets you can say, "On the open source box for this test, I want to run it against all of the BIND 9.6s, all of the BIND 8.4's, just because we like history, and all of the NSDs." So when you run this project, the first thing they'll run is

BIND 9.60 and then BIND 9.61A. It'll just iteratively go through all of them.

Here's another example. This is if you're going out to recursors, and this is showing that you can in fact, as you can tell by the address – the Windows boxes – are local. Those of course don't come with the package because they aren't open source, but you can add your own. And if you want to check the recursors, you can go out to the Google DNS. So you can send things off-box as well. Same queries hopefully getting similar answers, but this is a way of comparing all of those.

The program that runs on the LINUX host takes the actions we're starting up and shutting down, and each time you start it up, it will send off the query and things like that. The queries can dig or getdns. In fact, there are examples of getdns in here because you might, for example, if you're doing DNSSEC, you might want to look at all of the records that came back, not just the "Was it positive or negative?"

One of the things that Willem didn't show was that getdns allows you to say, "Show me the end client all of the records I would need." Now, you would hope that that set would be the same each time, but in fact, with various broken middle boxes, you might not get them all, and that would be interesting.

It actually runs surprisingly quickly. If you're doing a very simple test like, "Just make sure that all of these authoritative servers know how to respond to a specific new record type," even running on a really old, slow laptop – which I got an old one from my dad and he wanted to erase the hard disk, so I put DNSHarness on it instead – the start-up,

shut-down for 350 authoritative servers, the whole run took less than three minutes. So it really does actually do it very quickly in the VM.

So again, the programs that run on the VM, you start up any of the servers that you want. You can do various things. It's not just "Start up a zone sever." For example, you could do something where you say, "Start up the server and roll a key." See how that reacts. See how those delays react. Tear it down. You might want to capture the program and such like that.

So this is just a real brief overview and the status is that it's been released for a while. We did the first release a couple years ago. DNSHarness.org is where you can go get it. It's all BSD-licensed, so you can pull it down. You can do lots of improvements. I did the work, but it was funded by Verisign Labs. I'm not under contract with them right now. I'm interested in doing stuff.

You'll find other tools in there you might find useful. There's a whole set of ways of sending out a purposely broken request or sending back purposely broken answers. Roy at Nominet had a system that does that. This is another one.

So catch me this week if you want a demo. I can certainly do that. Or if you have questions. Or if you just want to grab it later if you're a researcher, just go to DNSHarness.org and you should find it there.

DAN YORK: Thank you, Paul. Question for the group: how many people have used DNSHarness right now? Just curious.

PAUL HOFFMAN: Probably few.

DAN YORK: Probably few, okay. How many people are interested in using it now that you've seen what Paul is talking about? All right, good. All right.

PAUL HOFFMAN: Okay. Yeah, good. Thank you. Again, if you want a demo, just grab me. I actually have it running on a remote box. We can take a look.

DAN YORK: Yeah, this is great work. Any questions for Paul? Okay, thank you very much, Paul.

Now we're going to go into our two demos, and I think we're going to switch – oh what?

UNIDENTIIFIED MALE: [inaudible]

DAN YORK: Okay. So for the people who are remote, we are about to switch to a demo from Iain Learmonth about the DANE-enhanced OTR messaging, and we do have some slides which we'll be showing to you in there while Iain's going to be doing a demo live here in the room. So if we're slightly out of sync with what you're seeing, that's because we're watching the demo that's going on here.

IAIN LEARMONTH: Okay. Hi, I'm Iain Learmonth. I'm wearing my University of Aberdeen hat today. I'm going to do a quick demo of a tool that I developed at the next web conference at the Hack Battle hack-a-thon using the getDNS API, which Willem has introduced.

So DNSKEYS the name of the python library we put together, and it fetches fingerprints for verifying cryptographic keys from DNS. A lot of you are probably familiar with the TLSA record for verifying SSL certificates using DNS without CA. There's also drafts available for verifying open PGP keys, for e-mail addresses, and verifying off the record fingerprints. Willem mentioned SSH fingerprints can also be published. They're meant to be on that list, but I forgot them.

We also looked at building this into an actual jabber client – jabber the XMPP instant messaging protocol – which uses off the record encryption. So you've got keys. You've got fingerprints. If you want to talk to someone, you want to know that you're talking to the right person. Instead of using a certificate authority, you manually swap fingerprints. This is not an easy exercise. You might trust the phone network and trust that you recognize someone's voice and verify it out of bound that way. You might have to meet up in person and manually check the digits. It's not fun.

So we'd like to give away a bootstrap in this using DNSSEC. So if you publish your fingerprint in DNS, then when you come to verify the fingerprint, it should be able to give you a hint as to whether not you can trust it.

Hopefully, Willem is now going to start a conversation with me. There we go. So he started a secure OTR connection. This is a layer on top of the XMPP session, but I can see here that it's unauthenticated.

Now, Willem's published his fingerprint in DNS, so when I now go to verify it – this one – so I can see I'm in an encrypted session. I can see the fingerprints that he's claiming to belong to him, and when I go to verify it, I can see this fingerprint has been verified using DNSSEC.

So I can now say that I've verified that that is correct. Now if we refresh this session – there we go – it is now an authenticated session. I'm now sure that is definitely Willem that I'm speaking to.

If I go and try to verify, he definitely won't have – yes. We can see here verification using DNSSEC has failed. This should probably be in bright red and set off sirens to make sure that you don't miss it, but we can see the hint is there. I can still verify the fingerprint manually, but he's not fooling me.

For those of you who are viewing remotely, you'll have screenshots of the success screen and the failure screen. All of the source code is available online at this URL.

I'd like to thank the University of Aberdeen for paying for me to travel here, and Verisign Labs, and NLNet Labs for paying for me to go to Amsterdam to work on this. That's all I've got.

DAN YORK: Very cool. Any questions for this? And we did have a live demo that worked here. That was good. All right. Congratulations on living on the

edge there. All right. Any questions for Iain or for Willem, I guess since they're both up here?

WILLEM TOOROP: So nobody could see our conversation?

DAN YORK: Yeah, it was completely secured except being displayed on a big screen in front of 120 people or something like that. But yeah, okay.

Thank you very much. This is very interesting work to see that. I'd heard and I'd seen Paul Wouters draft about doing this, which I know you had briefly up there, and for people who are interested in more info, the slide Iain had that said referenced a couple of drafts are – yeah, those right there – draft, Wouters, DANE, open PGP and the OTR of P are ones that Paul Wouters has written and submitted – what?

UNIDENTIFIED MALE: Wouters.

DAN YORK: Wouters. All right. Sorry. Sorry, Paul, if you're listening. But he has submitted those into the DANE Working Group inside of IETF where they are being discussed at some point.

So thank you for showing some actual implementation here. Any more questions? Okay. Thank you very much, guys.

We're going to do a switch now to bring in Joost from – what is it? Joost? All right, I'm not going to do Dutch – yeah. Anytime there's Dutch names…

UNIDENTIFIED MALE: You just lost your ability to go to Amsterdam.

DAN YORK: Yeah. It's that – never mind. My Dutch friends have tried to get me to pronounce things in Dutch, and I just have an absolute failure to do so.

So, Joost – did I get that right?

JOOST VAN DIJK: Joost is excellent.

DAN YORK: Joost. Okay. Joost is here to talk about a demo, which he will do here. Folks who are remote, Julie has some slides that Joost provided, and Roland I see your name on the side. I gave up a long time ago trying to pronounce your last name.

Over to you, Joost.

JOOST VAN DIJK: Yeah. While I'm trying to set up my screen, let me just introduce what I'm trying to show in the demo.

I work for SURFnet, which is like a net service provider for Dutch universities. We offer different services to our universities. One of them is a DNS service. We have a DNS portal where people can manage their domains, and we have a certificate service where our customers can apply for certificates.

Last year, we integrated DNSSEC into the DNS portal, so our customers can now just, well, use a tick box to enable DNSSEC, which is nice because then that's all they need to do. It's very important for our customers because they're not DNSSEC experts. So it should be very easy for them to deploy.

This is actually very successful. There are a lot of domains that have been using DNSSEC.

Okay, this is the button I was looking for. So that's about it for the introduction.

Now, because we're actually a reseller of Komodo, and of course we all know about [inaudible] Dutch company once. We're actually very concerned about anything happening to our certificate service. So we're very interesting in new technologies like certificate transparency and also DANE.

We've been experimenting with DANE and we've been running a pilot to actually test if DANE is something our customers would be happy to use. So we integrated our certificate service and our domain name service into something that you can easily deploy DANE with – at least that's the theory.

So the [pilots] are still running. Well, we have some preliminary results and I'd like to show you right now. But first a little introduction about the way that certificates and domains are handled at our customers.

It's very important to notice that usually not a single person that's responsible for DNS and servers are using certificates and the certificates themselves.

Usually there are three different roles within our customer organizations. One is the SysAdmin who just wants to install a server and have the server attach some certificate to it. There's the DNS manager that just adds the DNS zones. And there's the local registration authority, who's responsible for issuing certificates.

So the problem is if these roles are distributed over different persons, then these persons need to cooperate to safely deploy DANE.

Let's have a look at the workflow. Usually it starts with system administrator setting up a server, so obviously it needs to collaborate with the DNS manager to have the server name registered in DNS. Then it needs the certificate, so a key pair generated, a certificate signing request is sent to the local RA, who does some validation, issues the certificate, and then returns the certificate to the SysAdmin.

Okay, that's what usually happens. Now if we introduce DANE, then of course there also needs to be a DANE record in DNS, so that would mean that the SysAdmin has to contact the DNS manager again to have the TLSA records in its zone.

Okay. Well, of course, this can be improved, this workflow, and that's what I'll actually show in the demo. So there's actually two separate

services. This is the service for applying for certificates. The local RA just goes to a portal site and has very basic interface. It's asking for a certificate singing request, so let's just pick one in the right directory, preferably.

So I have a certificate singing request here. This is the thing being sent to the local RA by the system administrator. It's a very simple interface. You just paste the certificate signing request and specify what kind of certificate you'd like, and then you would…get internal server error?

JULIE HEDLUND: I wonder if any of these demos work. That'd be great.

JOOST VAN DIJK: Yeah.

UNIDENTIFIED MALE: I could do that.

JOOST VAN DIJK: It's very simple. It's the simplest way to get a server error.

DAN YORK: For those of who are remote, we just had an internal server error in the demo. That was why we laughed when you heard that.

JOOST VAN DIJK: Thank you for mentioning this. Obviously I tried it five minutes ago and it was working.

DAN YORK: We commend you on doing a live demo. You've got innumerable points with us.

JOOST VAN DIJK: I've got a very bad reputation doing demos. I never learn, I guess. Let's switch to the other portal and see if that one survives.

This is the DNS portal. Here the DNS administrator registers domain names at its zones. Let's have a particular zone here. It's called the [inaudible]. So the DNS administrator can enable DANE for this zone.

Let's say that I would like to protect the website of [inaudible], I can just enable DANE. What this does is that it gives access to the portal to someone with a different role. So I'm the role of DNS administrator, but there's a second role that you can delegate: the role of DANE administrator.

This means that you can actually enable the system administrator to handle his own DANE records. So that's what the DANE tap here is all about. Once I enable DANE for the zone, then the system administrator can actually log into this portal and register DANE records for this specific domain.

This is done by just adding a service. Here I can say, "I want to have TLSA record for TCP Port 443." Of course, I have a backup plan where I have a certificate. Usually this is what's supposed to come out of the RA portal,

but fortunately, I can just upload my certificate to the portal. I need to remember where I have it. Should be somewhere here.

The certificate will be uploaded to the portal and the information in there will be extracted, and a TLSA record will be added to the zone.

By doing this, we prevent the workflow to be too complicated where all these people have to communicate out of [band], and now you have a system administrator who can update his own DANE records.

Okay. This is all fine. It's working. But there are some issues that came up with actually having users using these portals. It turns out that it doesn't work. It works in the most simplest cases, but it turns out that our customers already have problems using this workflow, so having certificate signing requests, registering TLSA records – so we can help them with these portals, but DANE makes it a little more complicated.

Of course, this is a really simple demo, but once you need to do a key rollover, things start to get complicated. The problem with the key rollover is probably well known here because the same problem is with DNSSEC. You should prepare your TLSA records before deploying your new certificate.

So if you apply for a new certificate, the system administrator needs to be very careful not to install the certificate before the new TLSA records are published by the DNS server. And actually, it has to wait a certain amount of time because it can be that the old TLSA records are still cached on resolvers out there.

So we try to automate this workflow with a certificate of rollover in mind, and I won't dare to show this in a demo, so I prepared some screenshots. So I do learn.

Unfortunately, I can't really size up this screen, but here you see the RA portal, and suppose that you are renewing a certificate because it's expiring soon. So you apply for a new certificate. What the RA portal will actually do is to check with the DNS portal to see if the TLSA records are published. If not, it prevents the system administrator from shooting himself in the foot by not releasing the certificate. So the portal will offer you the certificate once it is issued, but it will check if the TLSA records are there.

It will also wait for a certain amount of time, and we did the time to live of the old TLSA record to make sure that the old TLSA records are gone, and only then it will issue the certificate to the local RA, which can then be given to the system administrator.

So this sort of works, but there's some other problems. Let's switch back. Another problem is that the certificates are bound to a specific port number. That's what I showed you in the domain name portal. You had to put a port number in there because you can't have the different certificates on different ports on the same server.

The problem is this information is not contained in a certificate signing request. So you still need out of [band] communication between the local RA and the DNS manager because in DNS you have to specify the port number where you want the TLSA record for.

This makes for a very complicated workflow. So our preliminary results are not that good, because our customers are – I guess it's just too error-prone.

The good news is that we have been thinking about how to solve this. Well, we don't have the solution yet, but one of the solutions may be to have an integrated portal. So both of our portals can maybe expose some of the functions through an API and you can just have another portal specifically for DANE administrators to integrate the two separate services. In this portal, you could ask for port numbers and stuff, and then on the background, the original portals would be used to apply for the certificates and to edit the DNS zones.

Another solution may be to change some of the standards, and of course that's not very trivial to do. But I could imagine that there's an extra field in a certificate signing request – let's say an attribute – that would specify the port number for the certificate to sort of target the port number for the certificate.

Another thing we've been thinking about is to change the TLS protocol. Of course, this is even more complicated, but there's a recent extension called server name indication where you can actually ask for a specific certificate from the TLS server. This may make sense for DANE as well. So if the client that does the validation of the TLSA record would be able to maybe specific exactly which TLSA record it's interested in, or alternatively, if the server would just send all certificates associated with the server so during a certificate rollover it can both send the old certificate and the new certificate, this actually may be a solution to this deployment problem.

I notice that there's an operations guidance draft RFC on the DANE Working Group.

UNIDENTIFIED MALE: And you have author here. Really quickly, if you haven't read that, we're actually adding a lot to it right now. There should be a new version out. I don't know if my co-author published it yet or not, I'd love additional feedback if you have anything you think that we're missing in there. But one of the things we do say in it is you really, really, really should do SNI. You should do the server name indication because it makes a whole lot of things easier.

But we also talk about how to get new certificates, make sure you publish the new DANE record first, how to do algorithm rollovers, and all that kind of stuff. We stepped through that pretty carefully. So if you think we're missing something there, please do that as well.

JOOST VAN DIJK: Yeah. Well, I'm not sure if it's missing something. It's just that in practice with non-experts, it is really difficult to deploy.

UNIDENTIFIED MALE: Yeah. Like most security things, you're more secure once you get it working correctly. But security makes things harder originally. Then you're better until you need to roll or a key has expired. Then you're worse.

JOOST VAN DIJK: Yup.

DAN YORK: I see Roland and we also have a remote question.

ROLAND VAN RIJSWIJK-DEIJ: Can I briefly add that the main takeaway from our pilot project is that it's basically the work flows that don't align. You have an existing workflow for requesting certificates for processing them within an organization, and that does not align well with the DANE workflow where the rollover model requires you to have two records at the same time, whereas you typically replace your certificate instantly. You never have two certificates active. That's the biggest issue that we run into.

And people find it very hard to renew their certificate and replace it on time as is.

DAN YORK: That's true, and it actually gets even more complex if you have outsourced your website to something else, and they're going to swap it immediately, but yet the DANE record is in your zone and you've outsourced the SSL half to somebody else.

JULIE HEDLUND: We have a question from the chat rom. Mark Lampo, security consultant says, "We cannot see remotely, but did the presenter add a value for the certificate use field of TLSA? Hearing the demo, it seems a self-signed certificate was uploaded."

JOOST VAN DIJK: Actually we tried to make it as simple as possible. So the TLSA record that was published in the zone you need an exact match with the certificate, and it's using the SHA256 hash for it. So it was not a self-signed certificate. It was an actual certificate.

JULIE HEDLUND: Okay.

DAN YORK: Great. Well, anything else for Joost? Okay, we'll switch to our last presentation then. Thank you very much. Let's give a round of applause for all of our folks who are here.

Given that we're low on time, we'll just do our brief little last bit and then we will still be around here, so we will encourage you to come up and ask questions of folks that are here.

As we get ready, I would also like to commend all of our presenters. It's an interesting commentary that we heard no World Cup jokes this entire time. I'm surprised, especially with the Dutch around here. I was surprised. I was expecting one of you – Roland or somebody.

UNIDENTIFIED MALE: Someone should just go bite you from the neck.

DAN YORK: All right. So I want to say thank you to everyone who came here for being here for this whole day. It's always great to have these sessions, and it's great to see how many people continue to stay around here for this entire time.

Okay, we can go to the next – Julie's doing the two-handed thing. So we just want to end this by encouraging people to take some steps coming out of here about DNSSEC. We would encourage if you're a TLD operator or registry that we look at you to get your domain signed if you can.

To help automate that step of accepting DS records, getting more registrars, even with the 2013 RAA, we still have a good distance to go with getting more registrars making this easy for people to upload their records in there.

The other piece we're still looking at on the statistics side is we want to get down to that second level and start to understand more of who signed what.

Many TLDs – and I'm looking over at .se and .nl and others – all provide those statistics very easily. You've got our own stat sites. Interestingly, all of the new gTLDs have to put them into the CZDS (the Centralized Zone Database) and so you can actually get stats out of there and you can see there's a very small number that are actually signed, but you can actually see it.

We as a community and some of the folks working in this would like to see more of that, so anything you can do around that. Next slide.

For zone operators, if you're doing that, sign your zone. Work with your registrar to get some DNSSEC happening out there to work with that. Again, we're looking for statistics around that. Next slide.

If you're a network service provider, to the point that was made a couple times earlier, signing the domains is great, but we really do need to work on that other side as well and the validation side and we're trying to bring those numbers. I want that swoopy curve on Geoff's chart. I want to see a nice big increase. Yes, I want the hockey stick. Come on. I want to see that thing in there, so we want to get more validation happening out there. We'll see which can be a higher percentage. Well, we're already higher than Google's IPv6 chart, so we're doing well.

Anyway, also promote the support of the DANE protocol. We've talked about that here. We really want to help make that happen in some ways. Next slide, please.

On the website, if you're a content provider, again, we are encouraging, asking people, "Sign your zone." Look at that. Look about how you can get DNSSEC validating resolvers out there.

Also, one of the things that we consistently hear – and this is particular from the browser vendors – is they continue to tell us they're not hearing people ask them about DANE or DNSSEC support, so we are really trying to encourage people to ask about DANE and DNSSEC support so that if you have interactions with those folks – I look over at Mehmet and say, "Hey, Mehmet" – anyone who interacts with browser vendors in some way, if we can raise those questions about how can we

see about getting some DANE and DNSSEC out there. I'll go on to the next one.

For everyone, we of course encourage you to use DNSSECC in whatever ways you can. I also want to focus on the second one: sharing your lessons learned. The gentleman from HSBC who was here – I know he had to step out – but others, we do these workshops at every ICANN meeting. We'll have another one coming up in October in Los Angeles. We'll be putting out a call for presentations. If you've got an idea for a new tool, if you got a demo you'd like to show, if you've done some research, if you've got a case study of how you've implemented within your enterprise, we would love to be able to put that out to the larger community. These sessions are shown to the people here. They also are broadcast remotely.

We also record these, put them up on YouTube, and get them out there as well. So we're trying to get this out to the widest range. We would strongly encourage you, if you've got an idea, please. And you don't have to wait for the call for papers. You can contact me. I'm just York@isoc.org. Or you can contact one of the other members of the Program Committee that's around. We'd be very interested to talk to you about sessions that we could put up here.

I think with that, I would just like to say thank you on behalf of the Program Committee. Usually, Russ Mundy is sitting here with me doing this last part, but Russ has been sucked off into the ICANN NomCom process, which is why you haven't seen him here all day, but there are a couple of sites out there – DNSSECDeployment.org, the Internet

Society's Deploy360 portal and the DNSSEC Tools Project, all of which are projects by people who have been involved with programming this.

So thank you very much for your attendance, and I also want to give a huge round of thanks to Julie Hedlund again who has done an amazing amount of work to make this all happen in the background that you all don't see. But I could tell you she does a huge amount of work to make this all happen. So thank you all, and thank you, Julie.

For a change, we don't have to clear out of this room immediately, like we had to in past times, so you're welcome to stay around and talk to people. Talk to the folks who just presented or anything else. You're welcome to be here for a bit more. Thanks.

[END OF TRANSCRIPTION]