SINGAPORE – Tech Day
Monday, February 9, 2015 – 10:30 to 18:00
ICANN – Singapore, Singapore

UNIDENTIFIED MALE: This is Tech Day, from 10:30 to 18:00 on 2/9/2015 in Olivia.

EBERHARD LISSE: Good morning, everybody. There's a lot of hall in there. Can we maybe work on that? One, two, three. Can you hear me all well? Because for me, it's a bit of a hall effect, but it doesn't really matter.

Anyway, good morning. As usual, my name is Eberhard Lisse. I am the Chair of the Technical Working Group. Welcome to our 26th Tech Day. We have I think quite a good agenda today.

As usual, let me quickly go through with this. The first presentation will be from .ph. They have created a platform to connect different registries together, if I understand correctly, and Alan will explain this to us in a few minutes.

Then Cory Schruth from the IT team of ICANN will actually explain to us and show us what it takes technically to set up all these Wi-Fis when they always clearly don't work or they break down, so we'll know why and how much effort it is to get it working. It is usually working very well as far as I'm concerned. It's an enormous effort behind the scenes that we don't see. So we asked them to come and give us a bit of an idea of what they're doing.

Then Geoff Huston, who I haven't seen yet – there he is – will talk about routing in the new year. Simon Balthazar who I haven't seen yet – he's

there; thank you – will talk about their KSK algorithm rollover from .tz. Cristian Hesselman I've seen yesterday, so I know he knows when he has to speak. He'll give us a short update about the SECIR Communications Working Group the ccNSO has established.

Then – I forgot the first name –

LAWRENCE HUGHES: Lawrence.

EBERHARD LISSE: Lawrence Hughes is here. Sorry for that. He'll talk about identity registration protocol. Nigel Roberts, who is in the first row, will talk to us about securing small registries with IPSec. Then we'll have lunch. The IT team from ICANN was readily willing to have a practical session over lunch or have Josh Baulch available.

We have noticed that other groups, especially people from developing countries, have issues dialing into the Adobe Connect or being dialed from ICANN. Sometimes it doesn't work. Adobe Connect has such a facility that you can not only listen but also speak. Maybe it's a good idea to offer this once in a while to those who might make use of it. If there is no demand, then they will break for lunch as well.

In the afternoon, Marc Blanchet will give us an update from the IETF. Is Marc here? But Jacques Latour is, so I'm quite sure –

UNIDENTIFIED MALE: Marc [inaudible] room.

EBERHARD LISSE: Okay, no problem, but Jacques Latour sort of kept in touch, so I will see when he comes. Then Vicky Risk from ISC is going to talk about the decommissioning of the DLV. I don't know how many TLDs in the room make use of it, but it's good to know it's going to go away, and maybe it's good occasion that they talk about it. Then the host presentation from .sg is going to speak about business intelligence that they have developed.

Then due to popular demand, we'll have a coffee break, and the person who asked is not smiling – oh, he is smiling. Actually, the reason why we can have a coffee break is because the next presentation was developed by Luis Diego Espinoza and myself, where we're going to do a practical demonstration on how to sign a smart card HSM on the Mac. It's much easier than we thought it would be, so the demonstration will be a bit shorter than we thought it would be, so we reserved time than we actually need, which means Paul Wouters, if you're around, can start probably 20 minutes earlier, and then Jay can close at 5:00. I've changed the agenda slightly.

On the other hand, if we run a little bit late, it's not a problem. We have enough time to play with so I'm not going to cut any discussions or anything down.

So now I must go and change this to the other laptop. We are presenting from the Adobe Connect that sometimes gives a little bit of a fuzzy picture, but it's important for the remote participants that they also see, and if Kristina has to do something and she can't follow the

remote presentations manually, then the remote participants lose synchronization. So we're just going to have to switch the resolution.

Sometimes in technical meetings, we have technical problems. Adobe Connect is a bit reluctant today, so we have to load the presentation manually. We also have resolution issues, I think, so bear with us for a second.

EALDEN ESCANAN: Hello. Good morning. Thank you for coming to my talk. Yesterday we arrived in Singapore and checked into our hotel. There is this cloth, and then I said, okay, I wanted to open it so I could see through the window. When I opened it, it was just a wall there.

Okay, wow. It's so big. There you go. This morning I'll be talking about what we call Sinag. It's our new registry platform in the Philippines.

A very brief introduction: I am Ealden Escanan. I work with the Platforms Team of .ph. This is my e-mail address. Sinag is actually a Filipino word for "ray of light." The .ph logo has sun beams coming out, so I figured it's best to call it that way. Also, all the other names weren't really good.

Essentially, we wanted to offer EPP to our registrars. Unfortunately, our registrars had different development schedules, and we have a couple of small registries who wouldn't really switch to EPP. They would rather stay with a web-based interface and continue from there.

Also, we knew that we didn't want to implement our own EPP interface because there's already some differences in connecting. So if we make

use of an existing registry provider, it would be much easier for our registrars to connect, probably because they already have an interface, although at that point we're not really sure which one we should go to going forward.

We developed this system to be able to connect to different registries and then actually work with them simultaneously if necessary. Then we made it flexible by introducing a REST/JSON API and then integrated it with our existing platform.

Then we figured, "Let's make it open source, because probably some other ccTLDs will find this valuable." Also if forces us to do things properly because it's out in the open and pretty much there.

On a high level, this is our diagram at this point. On your left, we have different registries – CoCCA, FRED, etc. – connecting to our platform. Then we also have what we call the Partner Console, which is the web interface for EPP registries and non-EPP registries, as well as the Helpdesk team. At the bottom right, we have our old platform. So we're actually running our existing Legacy platform with our new Sinag platform together to generate the zone.

Let me talk about a bit more of our Partner Console. One of the things that we've been spoiled with, with our existing system is the reports, because we wanted to get different metrics from our performances registry. This is something that we carry over to Sinag with Partner Console. This is a portal for EPP and non-EPP, as well as Helpdesk teams to be able to register and renew, review registrar and registry performance, add and manage registrars top up, and then view

balances and transaction history. We wanted to make it mobile-ready as well, so it should work with your mobile devices.

For a sample of the screens, we have Earnings Report, pretty much revenue going to the registry. This can be drilled down per month, so something like this. The figures are not actual for obvious reasons. Those are very good numbers.

We also have Sales Report, seeing how much registrations and renewals we're getting, as well as in terms of years and in connection to the earnings report, which again we can drill down to specifically domains that we've registered for a period of time.

Performance Report is in terms of retention. How good are we in keeping our domains, as well as gain/loss? How good is our registrations from our renewals?

Snapshot just gives us an idea per TLD on the performance of our domains. Finally, we have breakdown for countries and which countries we want to target for our efforts.

We wanted to keep this flexible, again, to connect with the existing system. All connections actually go through our REST/JSON interface, so if ever we need to actually extend it, it can go through our interface, pretty much. And it's open source, again, or us to be able to write this properly, and it might be useful for other ccTLDs.

We're licensing it under Apache 2.0. It's pretty much built in Ruby and Rails, and then we are doing continuous delivery. Hopefully we roll out a new, small feature every week into our platform.

It's hosted in GitHub and since we are using it, there's going to be continuous investment from .ph.

The next two slides are a bit technical. If you go through the source code, you would see something like this. This is what you all a feature file, describing the features in English. This ties up with the code base, so it should guide development. Then we have automated tests around platform.

Going forward, this is an ongoing thing for us. We continuously enhance it. We're looking at more report details or drill downs going forward, depending on what our support and marketing teams need, probably more connectors and probably more payment gateways.

We also want to focus more on support tools for the Helpdesk team do they don't have to go through the development to be able to get something from our platform.

Finally, while we do have a REST/JSON API exposed, we still want to improve on it so that we can actually offer it as an alternative for our clients.

That's the end of my talk.

EBERHARD LISSE: Okay. Thank you very much.

EALDEN ESCANAN: Questions?

EBERHARD LISSE: Are there any questions from the floor? I see the first individual coming. Please identify yourself for the remote participants.

DAN YORK: Hi. It's Dan York with the Internet Society. Two questions. One, this sounds very interesting. Where can we find more information about the open source implementation? Where is it on GitHub, or where's the webpage about it?

EALDEN ESCANAN: Okay, so we're actually setting up the webpage. It's going to be at Sinag.dot.ph, and GitHub is going to be in our GitHub profile, .ph. Not all of them are up yet. We're still making sure that when we do publish it, we publish information that we can publish. You can actually reach me through my e-mail if you want.

DAN YORK: Okay. The second question related to that is, what do you do, or do you do anything, with DNSSEC as far as enabling registrars who might use this system to help with the exchange of DS records and things like that?

EALDEN ESCANAN: Currently we're not doing anything yet, but those were flagged to me in the past quarter, so hopefully we can roll it out in the next quarter.

DAN YORK: Okay. That is one of the challenges we've seen in looking at getting DNSSEC more widely deployed in areas: registrars and their systems such as this not necessarily having the user interface to help with the transmission of DS records. So that would be great if you could add that in there. Thank you. Thank you for making this available.

EBERHARD LISSE: Any other questions from the floor?

MARTIN LEVY: Hi. Martin Levy from CloudFlare. Can you just repeat your part about not using EPP? Did you have EPP in your system and then no one was using it, or did you never go down that route at all?

EALDEN ESCANAN: Okay, that's a good question. We never did go implement our own. Early on, we decided not to roll out because even the existing solutions are slightly different from each other and we didn't want to burden our registrars with that, so we decided to go on using an existing one.

EBERHARD LISSE: Stephen?

STEPHEN DEERHAKE: Stephen Deerhake, .as. In one of your slides, you were pointing to both a box host of some sort running CoCCA and another one running FRED. How do you handle the subtle differences in EPP specifications for FRED versus CoCCA, particularly with the name server stuff?

EALDEN ESCANAN: Okay, that's a good question. We have what we call a connector, which is unique per registry. So this connector is actually two ways. If we're pushing a change from Sinag to the specific registry, we go with a standard EPP. So per connector, it's actually different to work with the registry. If we're pulling data, we're actually pulling the database directly for it. So Sinag in itself doesn't know about the specific registry, but it knows about the connector, and then the connector is the one who knows the specific one. Not sure if I answered your question.

[ALEX MAYOVER]: I was wondering: Sinag manages a registry and you're interacting with other registries. Is that correct? So what is the type of data that flows between your Sinag system and, for example, the FRED system?

EALDEN ESCANAN: Sorry, can you please repeat your question?

[ALEX MAYOVER]: You are managing a registry with the Sinag system, yeah?

EALDEN ESCANAN: Right.

[ALEX MAYOVER]: So why does this registry need to interact with another registry system. Where is the authoritative data for the TLD located then? Is it in the

FRED system or the Sinag system? Or is it the registrar system what you are building?

EALDEN ESCANAN: In terms of the architecture, the authoritative part is Sinag, so actually there is duplication in information with CoCCA and FRED with Sinag, but the reports are generated from Sinag.

[ALEX MAYOVER]: Okay. Let's talk offline maybe.

EBERHARD LISSE: I have a question, too. How many registries are you operating if you say you needed to consolidate them? I mean, .ph is running on your Legacy platform, not on CoCCA, and not on FRED and other registry systems. I wonder why did you see the need to develop this? If you run CoCCA for example like we do, it can have EPP and the console. FRED has got EPP and there is a rudimentary console, but not really [inaudible] and other registry systems I don't even know about.

JOEL DISINI: I want to answer that, Ealden, if I can. Joel Disini with .ph. The reason is we wanted to move everything to CoCCA or FRED and just make a decision. We wanted to try something and see if it works, if the registrars have any problem connecting, if the registry is too slow. So we wanted a solution where we could try different registry softwares at a different time and see if there's a problem. There were commercial guys coming up to us saying, "We can connect to you, and we'll charge you a

dollar per domain or two bucks per domain, and we'll get you all these clients, all these registrars, to connect."

I'm thinking, "Okay. That's fine, but once I move over to your platform, I'm stuck. I'm stuck and it's very hard to get out." So what I want to do is I want to connect to them slowly and see how many clients, how many registrars, they bring in and see what kind of volumes they give us before deciding to move everything to that provider. So this gives you the flexibility to try different platforms, see what works, and see what fails.

Now, is CoCCA is able to handle a big volume of registrations? Our tests show that that's not the case. We see that FRED can handle a far bigger volume of registrations, but you won't know that until you've moved over and tried these things.

So we wanted to try different platforms, see what works, and then slowly decide which one we want to go with. That's why we wanted to work with simultaneous registries at the same time. If you're a registrar who's to using CoCCA, you can talk to us using CoCCA. If you're used to us using the FRED flavor of EPP, talk to us using FRED. If you want to use someone like a consolidator, like 1API, they talk to us, and we'll see what works.

At some point we might decide, "Okay, we want to use FRED, so we'll choose FRED." But we want to take our time switching over from our Legacy system to different registries. That's why we wrote the software.

We've been doing this for many years, and we figured what we want to do is go slowly into the EPP system by trying different solutions out.

Does that answer your question?

EBERHARD LISSE: Okay, thank you. All right, thank you very much. Give him a nice hand.

Let me have a quick look. Who's the next one? Okay, Cory Schruth is in the building. He will talk about how to set up an ICANN meeting from a technical perspective.

CORY SCHRUTH: Morning, everybody. While I'm loading this up here, my name is Cory Schruth. I'm the Meetings Technical Services Manager for ICANN.

Is that showing up okay in Adobe Connect for everybody, or are you seeing the notes also? All right, good to go.

A little bit about the Meetings Technical Services Department. We started in 2009, providing remote participation services. We started testing Adobe Connect during the Mexico City meeting.

In 2012, we decided to add some consistency and reliability to meetings services by dedicating some staff to our remote participation services. That kind of kept us pretty busy for a while, so in '13 and '14, we actually added a few more staff, so there are now three staff members that are completely dedicated to supporting ICANN meetings, both internal and external. So these sort of meetings, as well as Board meetings, SSAC meetings, registries and registrars, and pretty much anything related to ICANN.

Today we continue to support all these meetings services that you're using today, and in addition we also do a lot of research for future services that we may bring to other ICANN meetings. I believe Josh will actually be sharing a little bit with you during lunchtime for those who are interested about Adobe Connect and services related to that as well.

Like I said, we support a lot of these sort of meetings here. A good external meeting that we also support was the NETmundial meeting in Brazil back in April. That utilized a lot of new technology as well, in addition to the meeting that we have here.

Internet productivity: the most important thing that we provide is probably the Internet connectivity for the meetings because that's the foundation for all the other services to run on. We have to create a robust network that everybody can utilize for both wireless access as well as all of our services that run across that.

This is a picture of some switches in one of our shipping cases along with some access points and some Mac minis.

We utilize Adobe Connect, obviously, that everybody is using right now. It's one of our largest consumers of bandwidth at the ICANN meetings. We offer Adobe in both public and private meetings. Interesting thing: the more pods that we utilize here to send streaming video, audio, scribing and everything, the more bandwidth we consume, both from a user perspective, from a client perspective, when everybody's logged into the wireless network, as well as on a system level when we're encoding and sending that out to the Adobe Connect servers.

I'm just kind of going through some of the services that we are providing, Adobe Connect being the first one. The MP3 service: we do MP3 streaming and recording. We started to look at some of the off-the-shelf solutions to do this encoding, and we really couldn't find anything that fit with what we tried to do – multiple rooms, multiple languages per room, and everything streaming, encoding, and backing up to our off-site data center, all at the same time.

So we actually created a custom interface. We call it IStream, and you can see here that this is an example of one that was actually running this morning during our open ceremony tests. It has a full web-based control so we can hop into any room that we need to on the back screen there, and then the front screen allows us to stop and start recording, streaming, and actually see the VU meters to make sure that we're receiving audio. It's the same thing that's actually being run in the back of the room right here.

Another service that we help to provide is the real-time streaming, or scribing, that we've done in a number of meetings since 2010. This allows us to essentially send audio from the meeting location back to Los Angeles, where our transcriptionists are all sitting. They're typing out everything on their stenographs and then sending it back here via text. We utilize this service in up to three different meeting rooms all at the same time. It's actually pretty amazing how fast that all works. It's usually within about two to three seconds from the time that we speak the audio until it gets back here.

We provide presentation and cyber café laptops as well. There are presentation laptops in every meeting room, like this one sitting right in

front of me, as well as four or five cyber café laptops for public use, printing – things like that – up near registration.

One of the cool things that we've done in the past year starting at NETmundial in Brazil was offer the remote hub solution. This is actually a picture of the remote hub installation in Brazil there. From the left to the right, we've got some audio consoles that are especially for the remote hub audio mixing. We have video control on the left. Straight ahead we've got ten Mac minis hooked to ten different Skype accounts, all connected out to different parts of the world, and then all on the right hand side was our Adobe connect in seven different languages as well. So it actually took about 30 computers down there in Brazil to make all of the different languages connected into all the different Adobe Connect rooms and all the remote hubs.

The cool thing about the remote hubs is somebody is speaking, let's say in French, and we have other hubs listening in Spanish. That French speaker can speak, it goes through our translation equipment on site at the meeting location and then sends it back to the Spanish-speaking location. That all happens within a matter of seconds through potentially multiple interpreters to make that all happen in real time.

We utilize Voice over IP telephone calls to connect to telephone to the conference bridges so we can have people dial in from remote and connect up to the meetings in the event that they're not able to participate with Adobe Connect or another streaming platform.

We have the meeting registrations system that we support and operate, a custom-designed system for ICANN. I think pretty much everybody in the room has probably used it in order to get your badge. You go in

there, punch your e-mail address in, and it spits out a badge for you. If you're not previously registered, you can log in there on-site and set that up.

We also offer a meetings scheduling database. This is mostly for internal use, but some external constituencies us it in order to schedule their resources. For example, this meeting room I can see has a projector, has microphones and is an actual meeting room. So when requests are put into the database, it allows our meetings team to go in and massage all the data around and figure out what room is going to be best for what groups.

It's a new service that we've really pushed to implement for pretty much everything, including pop-up meetings over the last couple of meetings. It's done great wonders for keeping track of things. It used to be that we did all this on paper, and it was a big mess.

Digital signage systems: these are new within the last couple of meetings, where we have it dynamically linked to our scheduling system that we just talked about. It pulls in the schedules as well as other Flickr feeds, Twitter feeds, and information like that. You'll see these around the venue. I think we have five or six of them around. It allows people to walk up and kind of get current information on what is going on right now.

Power strips: everybody takes power for granted, right? We actually bring 450 power strips and lay about a mile of gaffer tape at every meeting to make the power at every table a success.

In order to get all that there, we have to ship it. We ship about 10,000 pounds' worth of gear to every ICANN meeting. That's everything from network gear to power strips to servers and some audio visuals systems. A lot of our audiovisual comes from the local venues – the speakers, the cabling and things like that, TVs – but pretty much everything else we bring with us because we can't guarantee that it's always going to be available at every single location that we go to. So we've found that in order to keep that consistency that I talked about earlier, it's important to pack it up and bring it with us.

We just got notified from our shipper a couple of days ago that we have overloaded a 20 ft. container, so we're now moving to a 40 ft. container on a ship across the ocean. So with every meeting, we seem to pick up more and more services that we're supporting. That's the list of stuff we just went through. The font size keeps shrinking in order to make everything fit on this slide, we've found out.

To make it happen, you've all seen me around. You've probably seen Josh around. We've got three other engineers, both supporting end-user support and remote participation services, and then we've got 16 support staff as well that contract with us for these meetings. This is all just to run remote participation services and make those connections in and out of the venue and set up the network. Pretty much anything that isn't audiovisual we do.

Now the fun part: actually deploying the network. It all starts about six months ahead of time when we do site surveys to make sure that the venues are actually going to be able to support an ICANN meeting.

Now, obviously this happens earlier if we can, but lately, especially because of the venue changes, these seem to happen pretty quick.

We'll actually go out to the venue. Luckily we've been here before, so we know what's in the walls and what it takes to put on an ICANN event.

We utilize a lot of the copper and fiber infrastructure that's in the venues. We have to come out on site and actually do a full test of the lines around the building between the MDF (the main distribution point) to the IDFs (intermediate distribution point), where all the switches actually reside that can run the individual meeting rooms. We'll usually do that between four and six months ahead of time.

At that point, we meet with the ISPs and contract with them for bringing in the bandwidth. We typically bring in two one-gigabit connections in order to support the meeting. That varies depending on location and attendance, obviously, but that's kind of our go-to number. I believe here we've got a one-gig connection from NTT, and the other connection is an 800 megabit from SingTel, because that's what they were able to provide to this location.

Then about two months ahead of time, we actually put the equipment on a ship and send it across the ocean so that it's here in time within a couple weeks for the actual meeting.

Then about nine or ten days ahead of time, depending on location, the crew starts traveling in. About a week before, we start deploying the equipment, and usually four to five days before the opening ceremony is when we can see the network come online. Wireless access points

start to be deployed, and we have small meetings starting to pop up that are actually using the services.

During the event, obviously there all the services that we talked about earlier. Then usually a couple days after the event ends, or after the Thursday meetings, we start tearing down the network, packing everything back up, and throwing it back on a plane or on a ship in this case.

In addition to all the large ICANN meetings, again, we support a lot of smaller meetings. In addition to all the equipment that fits in the 20 ft. container that might be sitting on top of the ocean, we've got a lot of other equipment based in our offices that can support some of the smaller events.

In order to bring all that together, here's what the network actually looks like. Like I said, we bring in two Internet feeds and run it into two Juniper routers that are cross-connected to a virtual chassis, an EX4550 Juniper switch. Then we're utilizing an Aruba wireless controller, two of them in a cluster.

One of the new things that we've done for this meeting, which so far has been a good success, is we're using lag groups, or multiple links to each distribution switch. It seems to have reduced the amount of outages due to bad cabling, bad fiber optics, bad optics – anything like that. So everything from our core through to pretty much the IDF locations right outside of each room has at least one. I believe we're actually using four for the main ballroom, just to make sure that that doesn't go down.

Breaking out into the meeting rooms, some meeting rooms have multiple links back to distribution switches, and other ones just have a single link, depending on the cabling infrastructure that's available throughout the venue.

This is what the core actually looks like. I just went through all the equipment. It's sitting in a room about two rooms over from here. And that's about what it is.

The IDF locations, like I said, we cross-connect all those to all the individual meeting rooms, and those uplink to the main core switch back in the MDF.

Sometimes we get lucky. This is a picture from Los Angeles. It looks pretty good. Sometimes not so much. A lot of our time spent for the site surveys in some locations it takes multiple site surveys. London I think I went there three times before we had the event to make sure that we knew what wires went where. There was unfortunately a lot of lack of documentation, and that seems to be very, very common almost everywhere we go.

This place has been very nice. Every port's labeled. Every link is labeled. They ran brand-new fibers for us before the meeting last year. So we've been very, very lucky at this location.

Some of the other locations have been tough, and it does take a lot of trial and error to make things work.

Some of the dependencies we have for services that we don't have a lot of control over: Adobe Connect is a great example. They're doing a maintenance tomorrow, 10 AM to noon I believe, local time here in

Singapore. If we're using Adobe Connect tomorrow, it may drop. Unfortunately, we've tried to contact them and get this resolved, get them to delay it, but we're not a big enough customer for them. So if anybody has any contacts at Adobe, I'd love to talk to you after the event.

But these are some of the dependencies that we rely on in order to pull off an ICANN meeting. Pretty standard list there.

As for systems monitoring, we have a projector in our NOC, which is in the Enterprise room. Everybody is invited to swing by and see how it all works.

It looks something like this. We load in a map of the venue. We utilize a program called InterMapper and then place essentially ICMP or SNMP monitoring probes on the map about where the devices are located. If everything's green, like it was last night, we're all good. We have it linked out so it makes an audible noise in the NOC as well as on our laptops if something goes down, something gets unplugged, or there's a configuration issue and it goes offline.

We also keep track of some of the network utilization, both on the wireless and the wired. As we're building up the network, obviously more and more clients are joining. I saw about 1,100 this morning, right after the opening ceremony. We took this this morning before things got started.

Actual bandwidth utilization on the wireless hasn't been so much. We usually see that peak out Tuesday and Wednesday. Those would be the

high days. We usually utilize about 250 megabit on the wireless during peak times.

So far, we've only seen about 150 meg between the two routers in the two different ISPs, and that is combined IPv4 and IPv6 there in that view.

Is Michele in here? He asked for some v6 stats. I looked at it last night, and we're actually running almost 9% of our traffic via IPv6.

That's about all I've got right now. Questions?

EBERHARD LISSE: Nigel? And then Elmar.

NIGEL ROBERTS: First of all, I'd like to say thank you. This was one of the most interesting and directly useful presentations I've seen in the whole of ICANN for a while. Thank you very much for coming.

CORY SCHRUTH: Thank you. I appreciate it.

NIGEL ROBERTS: Just another little comment, really. You talked about the scribing and you said how fast it is. You do realize of course that everybody who knows basic physics knows that two seconds is physics. It's the speed of light that you're constrained by here, so it's actually a lot faster than you think.

CORY SCHRUTH: Yup. We are about 200 milliseconds away from where the scribes sit, so it is very fast, but it does take a little bit of time for them to type it. They're incredible. They're very fast at what they do.

ELMAR KNIPP: I also want to start with a thank you for the presentation, and also for the great work that you're doing. It's wonderful.

CORY SCHRUTH: Thank you.

ELMAR KNIPP: My question is, can you give us an overview about the cost or the budget you have per year or something in estimation?

CORY SCHRUTH: Honestly, it varies, depending on location that we go to. Also, it ultimately depends on the cost or the sponsorship of the bandwidth. I would have to pull up some documents on the actual budget for the services that we contract out and the staffing that we utilize, but the actual operation during the meeting Internet-wise varies on location.

JIM MARTIN: Morning.

CORY SCHRUTH: Hi.

JIM MARTIN: I'm Jim Martin from ISC. Just one quick question. I notice that you're NATing on all of your networks. Is that an intentional goal to degrade your v4 service so you got a better v6 service?

CORY SCHRUTH: Honestly, we used to utilize public IP space during the ICANN meetings until I want to say about five years ago, and we were trying to do the right thing and contribute v4 address space back to the community. So we actually released some of our v4 address space. By doing that, we had to NAT, and for most cases – I'm not as technical as you are – it seems to work fairly well for normal, general browsing and operations like that.

Obviously there are things we've heard this week and at prior meetings of VPNs not working correctly, and we're trying to work through those. There's obviously some tricks that we can do to make that work a little bit better, but obviously I'm more than willing to discuss what we can do. Maybe for the people that require public IP space, we can talk about setting up either a separate SSID or reservations or something like that to make sure that you get the requirements that you need.

JIM MARTIN: No, honestly I'm thinking that perhaps forcing people to v6 if they want real connectivity is a win. That's interesting I think.

CORY SCHRUTH: Yeah. We are distributing v6 DNS servers now. The new Junos 14 is allowing us to finally do that from the Juniper routers. So that's available now. I actually went v6-only for about an hour the other day.

STEPHEN DEERHAKE: Stephen Deerhake, .as. I also want to thank you, both for the effort and the end product here.

CORY SCHRUTH: Thank you.

STEPHEN DEERHAKE: It's astounding. I was wondering if you are aware of any other group that is doing anything near what you guys are doing – I'm thinking APRICOT off the top of my head – and whether you share this engineering expertise or whether this could be a profit center for ICANN going forward.

CORY SCHRUTH: We've definitely looked and talked with other groups, and we have worked with groups like ISOC to kind of share knowledge. We could do more to make that happen, but there's definitely some opportunities to kind of work together and consolidate our resources and make sure that all groups are out there being able to utilize that knowledge that we've designed and built. We have a lot of custom applications that I think could potentially be used by others.

SLOBODAN MARKOVIC: Hello. I'm Slobodan from the .rs registry. Thank you for the really interesting presentation.

CORY SCHRUTH: Thank you.

SLOBODAN MARKOVIC: I'm wondering if you could share with us a bit about your experiences with remote participation and organizing it and making remote participation more quality in terms of engaging people or giving them the opportunity to perhaps participate in their own languages.

In terms of the platforms and the technical capabilities, what challenges do you see with, for instance, enabling someone to, I don't know, type in a question in Arabic and have this translated and wiring all of that through the infrastructure?

CORY SCHRUTH: Yeah, we've done this. NETmundial is a great example of this, where we had the ability for people to chat in their own native language. We had translators on site as well as interpreters. The interpreters were doing the audible interpretation, and then the translators were actually translating questions that were coming in in the chat. It took a team of I want to say 30 interpreters and translators to do that portion of it.

We've looked at other solutions, such as automatic translation through, say, Google Translate or something along those lines. It's almost there. It's no quite there. In fact, Adobe Connect has a translation module or a pod that can do some of that automatic translation, and we've tried it

with the ALAC group for doing some of that. But it's not quite primetime yet, so very soon we'll be able to do something like that.

SLOBODAN MARKOVIC: Okay, thank you.

CORY SCHRUTH: Yeah. But just so you know, we are actively working on that.

SLOBODAN MARKOVIC: Oh, yes. A colleague of mine is trying every time to get his name written in Cyrillic on the text, and it doesn't work, so every time he gets question marks. So perhaps you could allow people to basically help us.

CORY SCHRUTH: Yes. That is a work in progress as well.

SLOBODAN MARKOVIC: Thank you.

EBERHARD LISSE: Okay, we'll take one more question.

NEIL EL HIMAM: My name is Neil El Himam from the .id registry. Thank you for the presentation. I have a quick questions. Is there any documentation with regard to the minimum requirement of having this type of service if you

propose to have a meeting, say for example, in our country? Do you publish that?

CORY SCHRUTH: Yeah, we do. As part of the requirements document for a meeting venue, we do publish the technical requirements as well. And they're actually very minimal. It requires power, the bandwidth requirements, and there's not a whole lot more. Maybe a little bit for the venue to support the meetings: the infrastructure, the cat 5 cabling, the fiber optics in and out of the buildings. But that's about it. It's very basic.

The nice thing about the system that we've designed is we're mobile. We're not asking you to provide equipment for the network and the remote participation operation. We can bring all that with us.

NEIL EL HIMAM: Okay, thank you.

CORY SCHRUTH: Yeah.

EBERHARD LISSE: But before you leave, Diego Espinoza has been involved from the NIC-CR side on the Costa Rica meetings, so maybe you should take this offline and make him give you some pointers.

I've been involved in looking at the ICANN meeting in Namibia, and, indeed, the requirements that the host has to provide are not that enormous. You have to basically have a fiber cable going in preferably

too and you need to have stable power. In developing countries, you need to have a good generator system if the power is not stable. Then you have a reasonably big enough venue. And for ALAC, you have to have proper hotels. Then it's okay.

The great thing about this – I'm really impressed – is that basically the first work more on technical infrastructure is transported on site, so if the country doesn't have all of this, it's not a big deal. The host really has to provide other basic things.

One more thing: yesterday – yesterday was it? – I happened to walk by their NOC. What room is it in again?

CORY SCHRUTH: Enterprise.

EBERHARD LISSE: Enterprise. It's really cool. If you find your way, make a turn there. It's just this big monitoring thing, seeing what's going on and how many people are registering or have registered. It's just impressive to see.

Okay, thank you very much.

CORY SCHRUTH: Thank you all very much.

EBERHARD LISSE: Next will be Geoff Huston.

Just one thing. On the agenda it says Routing in 2015, and the presentation is about routing in 2014.

GEOFF HUSTON: Hi. I'm Geoff Huston and I'm not a clairvoyant, so this is all about addressing and routing as it happened last year, not as it will happen this year.

I work in the regional Internet registry system. I actually work in the one that serves this area of the world, in the Asia-Pacific. What I want to do very quickly today is actually cover what happened in the addressing and routing system last year, and what can we learn from that.

On the left. This is not a current slide pack, but I'll go with it anyway.

Can I actually use a different pack? This is quite an early pack.

EBERHARD LISSE: Yeah? Kristina, can you help? He wants to share the screen, and I don't really know how to do it.

GEOFF HUSTON: Sorry about this, folks. I submitted this a very long time ago and I've got a much more succinct – and I think a much better – presentation pack sitting there.

EBERHARD LISSE: Is Cory still here? Because he forgot his USB stick.

KRISTINA NORDSTROM: We would have to open this first, I suppose.

GEOFF HUSTON: That would be a good thing.

Isn't technology wonderful?

EBERHARD LISSE: On the outside Adobe, it comes on – actually, I can see that.

GEOFF HUSTON: So I'm going to actually have a look at addressing and routing last year and sort of look at this in two parts. The first part is what happened to addresses last year and the second part will be what happened to the routing system.

You might have figured it out by now, and certainly the amount of NATs out there means that it's bloody obvious to anyone. What we've been predicting since 1989 finally happened. Addresses ran out. It's taken us a little bit of time to run out because the regional registry system didn't work the way I suppose we thought it would work 25-odd years ago. We kind of thought that exhaustion would happen from a single common pool, but as it happens, because we have five separate RIRs, and each of them operating from their own address pool, oddly enough this whole run out thing has been a long and protracted process.

The pool operated by IANA handed out its last addresses at the start of 2011. APNIC rapidly crunched through its final addresses that year and ran out in April of 2011.

The RIPE NCC did a much more orderly run out and ran out in September 2012. With LACNIC, there was much more evidence of a last-minute panic, and it ran out in June of last year.

We seriously expect ARIN to run out soon. You kind of go, "Can't you do better than that?" It's hard. Part of the issue with addressing is that we actually see two parts of this industry. A small number of you are enormous. You number your customers in units of millions. The rest of us are tiny.

The problem is, when a big person comes along and says, "I need addresses," they don't just want one or two. They want it in the millions. The pools we have now are numbered just in the millions. So predicting this is kind of difficult. It's not as orderly as we'd like.

So ARIN will run out. It will run out in the coming weeks or months. Precisely when? Difficult to tell. If you haven't requested addresses from ARIN yet and you're inside that region, things aren't looking good.

We can actually see what happened when we look at the last ten years of addresses that left the door for each RIR. This is the general look. What you see from 2005 to 2010 is an industry that was taking out the Internet more and more every year. In that last full year of operation, which was 2010, a little under 250 million addresses left the factory. We're not immune from normal business pressure, and that whole global financial crisis, when Lehman Brothers went to the wall – remember that and the year of hell thereafter – certainly affected the level of investment in infrastructure in the Internet as well. You'll notice that 2008 and very much 2009 reflect that downturn in investment.

In 2010 it kind of bounced right back up, and if the industry was doing a normal conventional roll forward, the inherent demand for addresses would be well over 300 million addresses – one-third of a billion – by now. So if we had an infinite address space and all that kind of nonsense, we'd actually see this industry expanding like crazy.

The impact of exhaustion is clearly evident. The bottom line (blue), APNIC, in the first four months of the year went through 120 million addresses and then went bang. In the next year RIPE in that final year went through 40 million addresses and then went bang. So what we have now left is just ARIN and AFRINIC still allocating addresses. LACNIC went bang last year.

So where did they all go? Obviously, there are two factors involved in where addresses go: the population and the GDP. If you combine the two and look at countries and GDP per capita and sort of weight the two, these numbers are not surprising.

Up until 2010, China and the U.S. were both taking heaps of address space. The U.S. doesn't have a very big population, but it does have a relatively high GDP. China has a relatively lower GDP per capita, but there's an awful lot of them, and that is a big GDP irrespective. So China and the U.S. were both pulling just under 50 million addresses per year.

The other countries in that top ten list – Japan, Australia, India, U.K., Germany, Russia and Brazil – kind of reflect developed economies working full speed on a digital economy. But you notice the next year as APNIC runs out, there are few folk who just appear for a year, like an allocation in Indonesia of seven million addresses, and the Republic of Korea knocked down to seven million.

But then APNIC runs out, and the year after what you see in 2012 is the entrance of some of the European entities claiming more addresses. Italy comes in, as well as Romania, and of course Russia.

The NCC drops out and in 2013 we're left basically with North and South American and Africa, and as LACNIC runs out in 2014, a much, much smaller address pool. The U.S.A. is still working at about 25 million addresses per year, but then Brazil at 10 million, and then Morocco and Columbia, which are relative newcomers coming out of AFRINIC, and down at the bottom of that list, Kenya and Mexico. So that's where they went.

Don't ask again, "What is this machine?" The demand for the industry? 300 million addresses per year. The supply side? Under 50 and falling. Demand is not equal to supply. What do the folk who need addresses do?

Well, you can't run a v6-only network; v6 is not a substitute for v4. The best you can do these days is run dual-stack. So there is still an inherent demand for an awful lot of v4 addresses, but unless you're in the U.S. (North America) and unless you're in Africa, your local regional registry can't really sustain that.

So what do you do? An awful lot of you are running more and more and deeper NATs. It's certainly true that around 95% of the Internet, particularly on the client side, sit behind at least one level of address translation, i.e., address sharing. But these days, it's more and more clear that we're seeing two and even three levels deep of address translation so that the original public address might be shared with

upwards of 10,000 discrete and different people at any point, which is amazing that the technology can withstand that.

But sometimes you can't share addresses, and in particular on some forms of secure transport, the certificates don't allow you to share addresses unless you're feeling awfully like trusting your neighbor with the most intimate of your secrets, which is a bad idea.

Sometimes there really is no substitute. Particularly if you're doing it on the service side, you need your own address. Where do you get them? Well, ultimately we're turning to markets. Some of those markets are clear to us. We can see sales and the registries register those transactions. So some parts of these sales are visible.

Others are less clear. They call them leases. What actually goes on is the address is transferred between two parties, but there is no clear public signature that the address has moved from one holder to the other. It's not necessarily visible to the address registry system.

So these two sets of figures are what we see. The first is the number of transactions/transfers per year. And as you notice as the registries run out from a small activity in 2012, a little under 200 transactions, 2014: 1,300 address sale transactions were processed across the three RIRs that are doing this.

If you look at the bottom one, the volume of addresses that were moved, it's gone from a little under seven million addresses, which is approximately half a slash-8 or a slash-9 if you're really into it, into a full slash-8, or the equivalent thereof last year, 16 million addresses were moved.

Are we moving old addresses or new addresses? Because the Internet itself, the address plan has addresses that are up to 25 years old. The red line is 2012; the green line, 2013; the blue line, last year.

It's kind of interesting that as we push further and further into transfers, we're looking for older an older addresses, which is kind of what the system was designed to do. So 6% of the addresses that moved last year through transfers are more than 20 years old. In other words, they were sitting in idle pools and were flushed out through money, which is theory and practice working together.

Interestingly, that period of five to ten years is the bulk, and in fact the most recent stuff, which is around the two to five year bracket, is the predominant market space for addresses. So between two and nine years old is what gets transferred.

I'm trying to align this. Okay. The number of v6 allocations per year. v6 is weird. v6 is weird because it's so big, and the big allocations are massive, while the small ones are hardly visible.

We view v6 in a couple of ways. This is the number of transactions, not the volume of addresses, per year. It's true to say that despite all the rhetoric about v6, prior to around 2009, no one really cared. The total system we were doing less than 1,000 of individual allocations per year. It really wasn't much.

You'll notice though that as the prospect of exhaustion came along in 2010 and then the reality in 2011, we started to see much more allocations of v6, and here there's no real evidence of the global financial crisis hitting those allocations. The folks who were doing this

were doing this on different funds, and my suspicion is it's not production money. It was research and development money. It's looking forward. Fascinatingly, since 2011, the allocation volumes have largely topped off.

What about by addresses? Well, that's a very different story, and part of the issue is, as I said before, there are massive allocations and there are tiny allocations. In 2013, that major block of red to ARIN was a single allocation into the U.S. defense department, so big that it just dominates everything else. That's why we kind of find that when we look at the volume of addresses, large and small don't correlate easily. So that volume can be quite misleading.

Where did they go? GDP plus population slightly different from the v4 table, but not by an awful lot. In this case last year, the United States got the equivalent of 4,900 slash-32s and China only half that.

Although interestingly a couple of the more recent allocations in China have gone towards massive – and I do mean massive – sensor-based networks. So we are seeing in China in particular the use of v6 in the Internet of Things. It's not so clear that in other countries it is quite such an overt link between the address allocation and a use in the Internet of Things in particular.

But then in other cases, predominantly Europe, the U.K., Germany, Netherlands, Russia, France, Italy, Switzerland with just Brazil as well. So in v6, the level of interest is predominantly northern hemisphere and predominantly around both side of the Atlantic more than anywhere else.

Now I'll get onto routing because what I wanted to do then is take that address data and see exactly how it gets applied. Routing is fascinating because routing is one of the few places where the entire Internet comes to your router. So if you analyze what you see, you're not just looking at your own neck of the woods; you're actually seeing a map of the world. You're actually seeing the entire Internet all at once.

This is the routing system since 1994. In other words the last, what, 20 years-odd of routing, and almost every major event is visible. So in 1994, we changed the routing system from an exponential growth to something slightly more linear. You can actually see the great Internet boom and bust of the year 2000 and slightly beyond that.

The real deployment of the Internet was in the noughties, and it really was broadband-based deployments between 2000 and 2010.

The global financial crisis is certainly evident there as well, but fascinatingly, address exhaustion isn't. That period from 2011 until now hasn't really changed the dynamic of the routing system.

When we look at the routing indicators for v4, and here's just a few of them, we're steadily adding about 45,000 prefixes every year. Haven't stopped, despite exhaustion. In terms of autonomous system numbers: 300,000 a year. It's like a machine.

The only place where I can see address exhaustion is with the amount of addresses that are actually advertised into the routing system. That's tailing off. Coupled with that is then granularity of address advertisements. They're getting finer and finer and finer. So instead of routing large chunks of address space, increasingly we're routing very

small bits. More Specifics] still take up half the routing table. That's just life.

But the other thing that is fascinating is in the ISP industry, no one wants to be a customer of anyone else. Everyone wants to connect as close to what [this is] the core of the Internet.

As the Internet grows, it does not get bigger. It does not get larger. It gets denser. So the average AS path length, which is down the bottom, has remained relatively constant that the diameter of the Internet is approximately ASs: three networks wide.

The one reason why the routing system originally designed for less than 10,000 entries, now coping with more than half a million entries, scaled so well is that behavior. As long as everyone nestles up tightly to each other, routing scales brilliantly. If we call decide we want to be long and stringy, say goodbye to routing. It'll die.

How do we get 45,000 new entries a year when we've run out of address space? This is a fascinating graph for me. I hope you find it so, too. This is the age of addresses that were added to the routing system each year, and that solid brown line is what happened last year: 18% of new addresses last year are more than 20 years old, yet less than 6% of addresses last year where registered in the registries that were more than 20 years old.

So what we're seeing is an increasing amount of the v4 address space in the routing system. The registries don't accurately reflect what's going on in terms of the address use, that as these old addresses come into play, not all of them are actually coming into play and registered

through the registry system. What we're seeing is the increasing use of leasing and other forms that have occlude the real address holder data from the registry system.

Also as you see that one-year point, more and more of the addresses that do get into the system are getting older and older. So one way we're doing 50,000 new entries a year is by recycling old addresses more and more.

The Internet isn't growing faster. That speed is coming off. Our current growth rate is 9% or 10% per year, and quite frankly it's because of this massive uncertainty between v4 and v6 that's causing us problems. We're really not sure to do with an industry investment that's slacking off. We're waiting.

And we're waiting for v6. Well, that's interesting. That's the v6 curve. That's the last eleven-odd years. Much steeper curves, isn't it? You can see when we exhausted, that curve got really big, and the next year we held World IPv6 Day, and I'm hard to say what happened after that, but growth is certainly coming on quite quickly.

But the numbers are smaller. Growth rate per year? Not 50,000 but 6,000. ASs per year? Not 3,000 but 1,600. Address consumption rate? It's not really fascinating. The average size of routing advertisements in v6 is actually getting smaller, not larger. More Specifics take up one-third of the table. That's just what we do.

Again, fascinatingly, the average AS path length is kicking around three. So what that really means is v6 is being routed the same way as v4, and that's a good thing.

Overall, growth rates in v6 are high, more than double that in v4, but previous growth rates were even higher. In 2012, it was almost 90%. It almost doubled. Last year, 2014 – the heading's wrong – a lot smaller.

If we're very lucky and continue with this phenomenal rate, we're going to be stuck with v4 for 16 years. You can't do that. You might be as clever as hell. You might think you can deploy NATs like crazy. There is no way you can do this. Something's going to break. So if you think we can just continue doing what we're doing piecemeal, you couldn't be more wrong if you tried.

Was that really the last slide? Or did I just go straight to the end? I think it was the last slide, but let me…

Sorry. Now it's working now. We're back again.

So how are we doing with v6? How much is there actually of v6 use? It is really weird because quite frankly the story with v6 is incredibly mixed.

This is a subtly different view of the number of users that use v6 measured every day. It looks a lot like the graph from Google, but I've spent a lot of time trying to analyze distributions. Part of the problem is, in certain countries, you get oversampling. In other countries, you get under-sampling. It is extremely hard to see what's happening in China, and there are an awful lot of Chinese users.

So it's not, as far I can see, 4% or 6% or anything like that. On weekdays, we get about 3.5% of the world using v6, and on weekends, less so, around 2.9%. Since Christmas, the numbers have come off. It's not a coincidence that the firewalls systems in China have been reworked, and it's certainly not a coincidence that the v6 numbers coming out of

China are dropping, not rising, and there are an awful lot of Chinese. One of the biggest influences on that slowdown actually comes from that country.

In the rest of the world, there are about 19 very large providers. Of them, some of the larger ones, like Comcast, are doing v6; inside the U.S., Comcast, Verizon, AT&T, and Time Warner; inside Belgium, a bunch of providers; inside Germany, the two large cable providers, [N-Deutsche and Telekom].

But if you look in other countries, like France, it's just been free. No one else does anything. In the U.K., the picture is disastrous. In fact, what you can say when you look at the world is that there are a small number of countries and a small number of providers that actually do something: Belgium, Germany, Luxembourg, U.S., Norway, Switzerland, Japan, and Peru where Telefonica are using Peru a laboratory.

But then you get down to, say, Singapore, par about 4%, and then a bunch of countries including Australia that are doing absolutely nothing. If you think you can stay like this, you're joking. It's not going to work like this anymore. Somehow, everyone else needs to get on board because we can't sustain what we're doing. The current world map of where we actually see v6 is remarkably spotty, and there's countries that are colored deeply red. It is a deep problem for the entire Internet.

Something has to change. I'm not sure what, but I think it actually involves you and I doing more than what we're doing right now, because if we don't, you can't survive on NATs for 16 years or more. It's not going to work like that. So somehow, something has to happen.

Thank you.

EBERHARD LISSE: Thank you very much. So it was about routing 2015 after all.

GEOFF HUSTON: One way or another.

EBERHARD LISSE: Okay. Any questions from the floor?

UNIDENTIFIED MALE: Geoff, can you go back a slide?

GEOFF HUSTON: I will. I'm going to try and see if I can get a better view of the world. That slide, or numbers?

UNIDENTIFIED MALE: No, the map.

GEOFF HUSTON: That's the map.

UNIDENTIFIED MALE: [inaudible] Western Sahara.

GEOFF HUSTON: Oh, it's really hard to get people watching YouTube in Western Sahara. It's just very hard to look inside that. I'm amazed that I can see inside other countries, but Western Sahara has eluded me.

UNIDENTIFIED MALE: [inaudible]

GEOFF HUSTON: Yeah.

UNIDENTIFIED MALE: [inaudible]

EBERHARD LISSE: Sorry, can you take this discussion on the microphone? The remote participants are also.

[ALEX MAYOVER]: When you said that IPv6 growth was slowing after 2012, do you see that NAT levels get deeper at the same time, which would indicate to me that people are getting lazy and saying things like, "It works like we're doing it right now"?

GEOFF HUSTON: There are three issues with v6 to make it work. Your machines need to have it in their stacks. So all the Apples and Macs and iPhones and everything else. Oddly enough, most of that is a solved problem. It

either is there or can be there. It's not that hard. Microsoft has been pushing out v6 ever since Windows whatever – Macs, etc.

The other part is the access providers need to offer it down the wire. That's an interesting problem because between the wire and the device is the home router, the CPE. If there's one dirty problem in this industry, it's crap-cheap CPE that users never upgrade, never change, and quite frankly, is worse than the analog television problem. No government is willing to give out free up-to-date CPE.

So we have this crap CPE that's full of open DNS resolvers as the basic platform for most of the world's DDOS, and at last count, there were 30 million of them, and no one seems to be able to solve it.

So we spend all of our efforts on v6, on trying to make the DNS work, and yet we're defeated by people buying shit product that really should be consigned to some digital bin. If we could change the world and get everyone to run up-to-date CPE, we'd find that getting v6 up into 40% and 50% would happen as a consequence.

But right now you see Comcast rolled it out U.S.-wide. Amount of v6 in Comcast? 35%. What's missing? Crap CPE.

That's the problem with our industry. We never quite figured out sort of between this and the network there's this dirty little box that you bought for 20 bucks. And it wasn't a bargain. It was a disaster. You should have spent more money.

[ALEX MAYOVER]: So essentially you're saying the service providers are not to blame. They are trying their best, but it's the consumers who are essentially [inaudible].

GEOFF HUSTON: Well, a long, long time ago back in Telco Land, Telco decided that providing customer with CPE was a waste of their capital. So in many markets, they said to the customer, they said, "Go and buy your own box." In the telephone world, maybe it was a nice thing to do, but in the Internet world, what came out was cheap product that is really, really old and badly designed, and nobody – not you, not me – upgrades them, so their functionality is basically broken.

Trying to get folk to actually change that is I suspect one of the biggest challenges we face today, because if we could get CPE working, you'd get rid of DDOS platforms. You could actually get v6 running. You'd get a whole lot of good things done. But I can't afford it. No government will afford it. As an ISP, no one's going to pay for it.

[ALEX MAYOVER]: Right. And to my personal experience, the CPE is probably the box that you replace most infrequently in your home, anyways.

GEOFF HUSTON: Absolutely. That's the whole problem.

EBERHARD LISSE: I'm going to close the queue after Dan York.

LUIS ESPINOZA: We have been talking about the implementation of IPv6 for years and years. The thing looks like it cannot be solved easily. From the perspective of the end user, he doesn't care about IPv6 or IPv4. He just needs the connection. From the perspective of the content, I think there's a lot of things to do. Not all the providers really have their content available on IPv6.

But I have been in a lot of LACNIC meetings, and we were talking a lot of times about the need of implementing IPv6 in ISPs, but I'm not sure what could be the solution. At the end, the ISPs will be moved because of what is there, what are the ends, and what will be the profit of this, and IPv6 looks like it's not providing profit for the ISPs. Maybe that's one of the reasons why they're not implementing IPv6 right now.

About the CPEs, I think the makers or the providers will build what the market demands. At the point that the ISPs really need or really demand good CPEs which would support IPv6, they will start building the CPEs. I'm not sure about what could be the solution of this.

GEOFF HUSTON: Common problems are difficult. We can talk about carbon dioxide in the atmosphere and climate change and the economics of changing things around. v6 has a very similar set of issues, where the cost is in one place and the benefits are at a different place and a different time – the benefits are to everybody, not just to the person incurring the cost.

So economically, it is a really challenging issue, and trying to think that the market will just solve it is kind of the bet each way. Look at Peru.

Telefonica Peru did something. And in the U.S., Comcast, Verizon, and AT&T did something. In Mexico, no one's done anything.

What prompts folk to do something? That's a tricky question, and I wish I knew the answer because we could poke the ones who aren't doing anything with the evidence of why others found it an entirely reasonable investment.

It's not impossible, but it's finding the right rationales and levers because you have evidence in some markets that they have managed to convince themselves it works. So in the South American context, maybe a conversation with Telefonica Peru might be very enlightening.

DAN YORK: It's Dan York from the Internet Society. I enjoyed your presentation as usual, but I just want to comment. I totally agree with you that CPE – and as you call it, Crap CPE – is a huge challenge we have with this.

But I do want to say I don't want to let service providers off the hook and say it's a consumer issue because I will say that my service provider, who I get my connectivity from, provides me with that piece of CPE box, and when I recently upgraded, I had to get a new CPE box from them.

Of course, the only they had in my local office that they would give to me was broken with IPv6. I had to do some digging through people who you and I both know to find out that it was broken and what was going on with it.

Now, they do give me a nice list of the devices I could go out and buy, but now I have to go and do the research to figure out which ones of those actually do work with IPv6, etc.

So it's a huge problem. They aren't easy challenges, but I will say in many places the service providers may be the ones who are actually giving those out, too.

GEOFF HUSTON: I heard of a very interesting initiative down this line coming out of New Zealand a couple of weeks ago. We all face massive problems with lousy CPE. We spend a huge amount of our time and money in security themes, trying to track down massive attacks, and it ends up being Belkin-based models with open DNS and all this kind of nonsense.

So they were saying, "If we all decided on open hardware, open WRT as the platform, and actually standardized inside New Zealand on a particular base of CPE equipment, could we fix this?"

It's certainly true that there is really good open hardware and software. Look at the Turris Project in the Czech Republic. It can be done and it can be very cheap and very effective. The amount of money you spend on putting the CPE in the user's home is saved by the amount of money you're not spending tracking down massive attacks and CPE-based issues.

So I'm not giving up. I think there's hope, and I think open WRT and open platforms is possibly the way out of this.

DAN YORK: Sure, and I agree with you. This is really one of these fundamental challenges we have. The challenge, though, even to that is, first of all, those of us who are in this room are probably interested in having good CPE to do it and will probably update them. But my parents won't. They bought a box that was cheap from Best Buy or whatever their provider might be, and the odds of them updating it are pretty much slim to none. So there is the challenge of, once you have the boxes out there, how do they get updated on a regular basis to fix those kinds of issues?

I agree. It's a massive problem that we have to sort out as an industry.

GEOFF HUSTON: Agreed.

DAN YORK: Thank you for continuing to do this and your metrics on an ongoing basis. It's good stuff.

GEOFF HUSTON: Thank you.

EBERHARD LISSE: Thank you very much, Geoff. The next one will be Simon Balthazar from .tz.

SIMON BALTHAZAR: Good morning. My name is Simon Balthazar. I'm working with TzNIC. I'm going to present on a project that we are currently working on. It's KSK

algorithm rollover. I'll be talking about the motives for why we're doing what we're doing, and the challenges that we faced, and how we are planning on solving them, in the hopes that someone who's going to go that route is going to benefit.

I have a bit of a background. The .tz zone was signed in 2012. That's when we had our DS published in the root zone. We chose the Algorithm 5 (RSASHA1) as the algorithm. The reason behind the choice of algorithm was based on the version of the registry software that we were using. It was only supporting one algorithm, and it's Algorithm 5, and we didn't want our registrars to interact with a registry direct on when they want to publish their DS to the zone. So the only choice we had was with Algorithm 5.

So in the quest of trying to do away with the current jobs and stuff like that, we decided to adopt a key management software, OpenDNSSEC, and we were using it for managing our keys.

In 2014, we did an upgrade of our registry software, and now it was capable of accommodating other algorithms in our registry. That was a good step for us.

In 2014, also the same year, we did our first KSK rollover. This was basically based on our policy that we roll our KSK after every two years.

The motivation behind this rollover basically was the usual ones: the deprecation of SHA1 algorithm. We wanted to do away with that algorithm that is almost exploitable, and having our registry back on track with an algorithm that is solid.

Also, the outcry of the technical community and .tz led to migration to NSEC3 instead of using NSEC to prevent zone walking because of the domains. So these are the motivations behind rolling our algorithm.

As we all know, the published steps for rolling a KSK are three, but in the quest of an algorithm roll, you cannot do the double DS because when you publish a simple of publication of the DNSKEY with a different algorithm, it will make a zone bogus. So the only two ways you can go with are the double signature or the double RR set.

So we had a couple of challenges, and basically there were only two, but they were a bit serious because the processes of managing the keys and everything at .tz is automated, and we were using OpenDNSSEC.

OpenDNSSEC posed two challenges that make it difficult for us to roll the key. One of them is that it cannot support multiple algorithm parts. We all know that the policy is binded on one zone. Also, the OpenDNSSEC currently doesn't support offline KSK. So we had to find a way of going around OpenDNSSEC and implement this rollover.

The only process was to go the manual process way, whereby we had to retrieve our keys from our current key management software and put it in a different machine and do these ten steps that we've highlighted on a different machine now, not on key management software.

I will not go through all of them, the ten of them, but to test these ten steps, we had to create a zone under .tz, and we call it nic.tz just for testing purposes. We use that zone to test each and every step to see if it was working perfectly.

The first step was to generate the signed zone with the double KSK and ZSK signatures using existing keys and the new keys that we've created. In this place, we had to make sure that the validity periods are more than long enough to cover any problems that we may encounter along the way.

Then in each and every step, we did a lot of testing on the machine. This was a live zone that was not being used, so we were able to do the tests even the tools that are online, like DNSVs and stuff like that.

After signing it and publishing the DS on .tz, this is what we had. We had a set of new keys that were Algorithm 10 and a set of keys that were Algorithm 5. After publishing that, we took the old algorithm out of the zone and remain with Algorithm 10. It worked perfectly fine. Like I said, this was done on a separate machine from our key management software, and the only task was now to move the new keys now into management software, which is the OpenDNSSEC.

We thought it would be a good thing to use the newer version of OpenDNSSEC, and we think by doing this process on a separate machine we can upload the new keys on the new OpenDNSSEC 1.4 that has just come out. It has some cooler features that we can actually use compared to the one we're currently using. So we create our OpenDNSSEC and go back to singing the zones again.

These basically are the steps you're going to follow in making sure the algorithm rollover is done perfectly.

Thank you.

EBERHARD LISSE: Thank you. Thank you very much, not only for presenting but also for tackling a relatively difficult problem. I remember talking to Jay Daley once when we said we might want to [haul] our algorithms and he just turned his eyes up because it's complicated and difficult. So I'm quite impressed that you went and did it. They're always saying there are only two ccTLDs in Africa that sign their zones intrinsically, and it's even more impressive that you do these things without relying on outside concerns and outside companies to do it for you.

Any questions from the floor? Okay, then let's give him a hand.

Cristian Hesselman is next. He will give us an update about the SECIR Working Group, the secure communication.

Can you just explain to us just shortly what SECIR stands for exactly again? I forgot.

CRISTIAN HESSELMAN: Yeah. It's on my first slide [inaudible]. Okay, thank you.

SECIR stands for Secure E-mail Communication for ccTLD Incident Response. It's a ccNSO working group. My name is Cristian Hesselman. I'm with SIDN, the registry for .nl, which is the Netherlands.

I'll be giving you a brief update of the status of the work in this particular working group.

How do we go to the next slide? Oh, here we go.

The objective of the working group is to implement the first version of ccTLD contact repository which stores contact information of people

that are responsible for the security and stability of their ccTLD, thus enabling the ccTLDs to quickly and easily obtain each other's contact details in the face of a large-scale incident on the Internet.

The other purpose of the ccTLD contact repository is to enable the ccTLD security and stability people to quickly exchange basic incident messages, messages on particular security incidents.

This is also why I'm giving this presentation here: I'm suspecting that a large part of the security and stability people will be in this technical community.

The other objective of the working group is to use a mailing list to implement version 1.0 of the contact repository. We're taking that approach because a mailing list is basically globally accessible and for everyone to use. It's relatively straightforward to implement. We can develop it at near-zero cost, which is basically a requirement that came out of the previous working group, the CRI Working Group, which held a questionnaire at the end of 2013. One of the requirements that came out of that questionnaire is that the system should be possible to develop and maintain the system at near-zero cost.

Finally, there are other similar initiatives at regional organizations, such as CENTR and LACTLD, and through mailing lists, we would be able to relatively easily interface with those initiatives.

The impact we expect from this working group, at least from the contact repository, is an improved handling of large-scale security and stability incidents that require a coordinated response of ccTLDs at the global

level. So it's basically in addition to the work that's being done by the regional organizations.

One thing to stress here is that the contact repository is not only for ccNSO members, but it's also explicitly open for non-ccNSO members.

The status of the work is that we have a mailing list up and running. The address is shown here on the slide, all the way at the top. Ironically, it's a list that was set up back in 2004 for similar purposes, but it had never been used. Or there were a few people on it, but I think the last e-mail exchange on the list was in 2010 or something like that.

So what we did was we purged the list and we basically reused it for this working group. The people that we are going to invite on the list are what we call the Security and Stability Contacts of ccTLDs. These are the folks responsible for the overall security and stability of their ccTLD.

We use the IANA administrative contact to authenticate or let them appoint the SSCs for their ccTLDs. So the IANA admin contact is basically our trust anchor, if you will.

Like I said, one of the requirements is that it should be possible to exchange rudimentary Internet information, which you can do of course on a mailing list. But we actually don't recommend it, so the main purpose of the mailing list is to use it as a contact repository and not really as a secure messaging service because the list is unencrypted at this point.

The administrator of the list is the ccNSO Secretariat, and as you already saw from the address, the list is being hosted by DNS-OARC, which is what we consider, let's say, a neutral ground.

This is the typical use of the list. There are basically three scenarios. At the top here, this is scenario one in which a security and stability contact can basically send a free format message or e-mail on the list asking for the contact details off another security and stability contact, saying, "Hey, I'm with ccTLD A and I want assistance from ccTLD B, and I need the contact information of their security and stability contact." Then they basically contact each other offline through a phone call or through instant messaging or whatever else. So that's the first use.

The second use is by sending a mailman. The mailing list is a mailman server. Mailman accepts special e-mail messages that contain a command, so you can send a command to the list asking the list who's actually on the list. This is something you do though the "WHO" e-mail message, and the list will then return basically a list of all current subscribers on the list. Then, again, the security and stability can contact each other out of band, though a phone call, for example.

The final use of the list is that we slightly modified the Mailman server in that it automatically generates messages once a month and it shares these messages on the list and the messages contain the current list of subscribers on the list. So it's, let's say, e-mail addresses, names, and mobile phone numbers in there. The purpose of that message is to basically remind people that the list is still active, so it's like a keep-alive message. Plus, everyone can store that message in their inbox for use, for offline usage, for example.

This is the procedure that we developed to get people on the lists. It's actually quite simple. What's happening is that the ccNSO Secretariat sends an e-mail to the IANA admin contact of a particular ccTLD to ask

them for the security and stability people of that ccTLD, which they then return in a reply e-mail message. Then they simply get subscribed to the list, and the Secretariat also sends an e-mail on the list saying that new people have been added to the list.

Finally, there is a step here in which the Secretariat updates a public directory on the ccNSO website which contains the names of the ccTLDs – so not the security and stability contacts – who are currently on the list.

The second type of subscription procedure is one that's not initiated by the Secretariat but by a security and stability contact who wants to join the list. It's actually pretty similar to the previous scenario, except that it starts here at the SSC and not at the Secretariat.

The status of the work is that we are currently subscribing the first batch of ccTLDs to the list. We currently have most of the ccTLDs in the working group on the list. That's Brazil, Canada, Denmark, Netherlands, and Tanzania. We have invited five more ccTLDs, so they're over there. We currently have Columbia and Japan – they should be green as well – on the list.

We're doing that because we would like to, first, evaluate the subscription procedure with the ccTLDs, and so far things have been going quite well. We have not seen any issues yet.

We also created a web page on the ccNSO site with more information about the list. It contains a description of the purpose of the list, who can be subscribed to it, and also how you can subscribe to the list and how to use it.

The next steps for the working group are to, in addition to those ten ccTLDs you saw on the previous slide, of course, invite the rest of the community. That will happen, let's say, after the Singapore meeting, so that will take a couple weeks to do that.

Then we intend to detail the interaction with similar lists at regional organizations. I also discussed that previously. Then we also will put up a plan for outreach activities and develop the roadmap for the working group.

Our end goal is to also include a secure messaging service in the platform, and in that case, a mailing list might not be the best option anymore. This is something for the long-term, so that's the roadmap we're going to be looking at.

And we'll be doing an evaluation – that will be around May roughly – and a write-up of our final report and the closing of the working group. We expect to be able to do that at the next ICANN meeting.

These are the folks on the working group. If you have any questions, I'll be glad to take them.

Thank you.

EBERHARD LISSE: Any questions from the floor? Thank you very much. The obvious problem is, in addition to getting people on the mailing list, is getting people to answer e-mails. That's one of the major problems that I find with many ccTLDs. If you want to talk, for example, to some of our neighbors, you send them e-mails [inaudible] they just don't answer

them. The e-mails are [validated]. This requires some activity from the ccTLDs that participate in this obviously. It does not work on just on just all [253] by default.

SIMON BALTHAZAR: Well, we're hoping that the mailing list will not be used because then there would be a large-scale incident somewhere, so it's basically a fallback facility. One of the things we're doing is that we're asking people to sign up with their full names and phone numbers and not use role-based accounts so that everyone can see who's actually on the list. Our guess is that this will enhance the trust between ccTLDs and thus will enable folks to use the list more easily.

EBERHARD LISSE: Okay, thank you very much.

SIMON BALTHAZAR: Thank you.

EBERHARD LISSE: Next will be Lawrence Hughes. We're running a little bit late, but I'm quite sure Nigel will be even more concise as usual.

LAWRENCE HUGHES: Okay. My name is Lawrence Hughes. I'm with a new startup called Sixscape Communications. What I'm talking about here is some interesting new work that I've been doing R&D on for almost a decade. Basically, I've got a lot of background in computer security and

cryptography. I created all the training at VeriSign back in 1998-2000. Then I spent the last decade trying to understand IPv6. I put a lot of money and time into mastering IPv6 and trying to brainstorm what we can actually do with it.

This is not a presentation on why we should switch to IPv6. It's what we can do with it once we have, which is a little different.

There we go.

First off, the IPv4 Internet is definitely broken. I'm not going to go into a lot of detail here. That's been done to death at many, many presentations.

Basically we realized in the mid-90s we were about to run out of addresses at the rate we were allocating them. There was no successor protocol at the time, so as a temporary measure – and this was stressed many times in the RFCs that it was temporary – we splintered the monolithic IPv4 address space into millions of private Internets, hiding each one behind one of the precious public IPv4 addresses using NAT.

This gave us another 10-15 years, but those years are over now. Even with these temporary measures, we have run out, as was covered so well in Geoff's speech this morning.

The successor protocol, IPv6, is now mature and being deployed globally. It's a little spotty in places, but it is finally happening. This is something that I'm now trying to figure out, "Well, what can we do with it now that it's actually happening?"

There we go.

So what are the main difference between the legacy IPv4 Internet and the new IPv6 Internet? Well, real address scopes; working scalable multicast, which is going to be a big thing in IPTV; more robust ICMP, now including IP address resolution; and autonomous address registration by nodes. But the two biggies really are ample global addresses, essentially an unlimited number, and no NAT. So let's finally celebrate that we don't have to put up with NAT anymore. This temporary measure is finally no longer needed.

What are the cool new things we can do with our shiny new NATless IPv6 Internet? For the first time since the mid-90s, any node can in theory – that's a big "in theory" – connect directly to any other node in the world.

We can leap past the client server architecture, which was made necessary by NAT, to Direct End2End connections. But we need better address resolution in order to make this possible. DNS is not really up to the challenge.

I started out by trying to create a secure End2End chat program that would allow me to go directly from my phone to anybody else's phone, and the first thing I ran to is DNS just really couldn't handle the address lookup.

DNS has no per-user authentication, so anyone can change registered addresses if they have access to the necessary [inaudible], or nobody can change it if you're not making that available. Neither one of those is adequate. You need per-user authentication.

DNS takes a long time for a new registration to be available to everyone. Propagation time can be 24-48 hours. That doesn't work very well in a mobile Internet where my phone might actually have six different IPv6 addresses during the day as I connect to different networks, get some of it from 4G, and maybe connect to my office network, my home network, and so on. By the time you find out my address, I'm no longer there. So we can't depend on an address resolution system. It takes a long time to propagate.

DNS is woefully insecure. It's very difficult to roll out DNSSEC on such a large existing critical infrastructure. There's something like 20 million DNS servers worldwide. I've got a lot of background in DNSSEC. I created the first DNSSEC server that was actually certified by JITC in the U.S. That's Joint Interoperability Testing Command of the DOD. Basically, I know what kind of problems people are running into trying to deploy it. The last talk was a good example of the kind of problems that people are having with the key management keys and so on.

DNS registers the address of nodes, and really static nodes, not of people, or more precisely, not where the person last logged in.

What are the benefits of being able to have devices connect directly to each other? There's no need for intermediary servers, no bottlenecks, or reliability and security issues from intermediary nodes. Traffic's decentralized. The traffic only goes over the shortest path between the two communicating nodes as opposed to out your ISP connection and then back in.

It's higher performance. If the two nodes happen to be in the same site, bandwidth is not limited by your ISP connection. It could be Gigabit.

There's nothing worse than having a Gigabit LAN and trying to send a file to somebody else in that same LAN and it has to go all the way up to Skype and back at ridiculously slow speeds.

Security is better. It's harder to monitor or block traffic than it is in the current client server model. When traffic is decentralized, it's really, really difficult to monitor or to block. You would have to have some kind of presence in essentially every LAN in the world to be able to monitor it at that point.

Overall system capacity is much higher, so a lot of benefits there. One obscure benefit is when I'm connecting directly to another node, I can use Diffie-Hellman for secure symmetric key exchange rather than using digital certificates. That's what's used in IPSec. It works very well, but it doesn't really work when there's multiple links involved.

What I did is created IRP. This is the Identity Registration Protocol. It allows a next-generation address registry. Any app can register the current IPv6 address of the log-in user at any time. It includes a user directory to provide pre-user authentication and linking the registered address to a person, not a node. Connections to the IRP server can be over IPv4 or IPv6, and they're over-explicit TLS, so only one port was required.

Authentication to the IRP server can be username/password when you first get going, but the first thing you typically do is request a client digital certificate, and thereafter it's cryptographic authentication.

IRP includes full PKI certificate management. You can request, download, revoke, renew certs, plus obtain other users' certs, check

certificate validity and revocation status, get CA certs, and so on. This allows hiding all the complexity of the PKI in the apps through the protocols, making it mostly invisible to the users.

One of the thing we ran into at VeriSign was they actually solved the problem with building sever certs pretty well, and a single PKI could actually handle providing all the server certificates for secure servers in the world, but there's many orders of magnitude more client certs. What they came up with really was inadequate for client certs, and in fact we never really sold any of it. So essentially, all the certs that VeriSign ever sold were sever certs, and yet client certs are what is needed for NDN security.

IRP is XML-based to make it really easy to implement and extend. Security issues: connections over IRP are secured with TLS 1.2, usually with X.509 certificate-based strong client authentication. These certs can actually be obtained from then IRP servers. In fact, I call the IRP server a Domain Identity Registry.

Once obtained, these client certs can also be used for website cryptographic authentication, S/MIME, network access, etc.

IRP provides all the necessary PKI for validity and revocation checking, so you don't need OCSP or CRLs or things like that. It's all built into the protocol.

Like DNSSEC, all the registered information is actually digitally signed on the server when submitted, and those signatures are delivered along with the information. So we're taking the concepts of DNSSEC but embedding them in this next-generation address registry.

Unlike DNS, the DNSSEC aspects were designed into the protocol right from the get-go. There's no complex transition after a massive infrastructure is already in place, so everybody will be having the DNSSEC-type aspects that's using it.

The key management for all this is built in because it's a distributed PKI, so there are no separate key management issues. It's all part of the overall system.

Okay, I actually submitted IRP to the IANA some time ago. It was reviewed by Lars Eggert, who is the Chair of the IRTF. It was determined to be viable and novel, which means it did not duplicate any existing IETF protocols, so the issued port 4604 for it.

I've actually created engineering prototypes of both the domain identity registry server and a first client that uses it, right about 60,000 lines of C# over the previous year. I got it up and running, verified that it's proof of concept, it's viable and it does actually work.

Sixscape Communications – that's my new startup here – is productizing these and will be creating additional products that leverage this.

As an example, we're making an add-in for Microsoft Outlook that'll make S/MIME e-mail completely trivial. It will automatically and go out and get the necessary certs. If you need to send an e-mail to someone else, you don't have to worry about them sending you a signed e-mail. It'll just go out to the IRP system and retrieve their cert and use it.

There will also be one for Microsoft Office that makes it really easy to sign, encrypt, and directly exchange documents with any other user. So a lot of possible things coming out of this.

We also created another protocol called a SixChat protocol, which was also submitted and came back with port 4605. This basically allows true End2End secure communications over IPv6 with no intermediary nodes. There is a retrofit version of it that allows one or both nodes to be IPv4, but that does require intermediary service. It's still secure End2End.

It includes a new peer-to-peer handshake that's equivalent to the one that's in TLS, using Diffie-Hellman for symmetric key exchange and then client certs for mutual strong authentication.

SSL/TLS is really hopelessly tied to the client/server architecture. It's not really applicable to things going directly End2End.

The SixChat application currently supports chat and S/MIME e-mail. Soon it will support file transfer, and later voice and video. You can think of it as kind of like a WhatsApp that's completely decentralized and has embedded PKI security with strong authentication and privacy and is military grade.

The SixChat agents depend on IRP for the address resolution in all the PKI aspects.

Okay, in case you're wondering what inspired our name and logo, if you look on the screen here, you might see what inspired us. Basically, our namesake organization, Netscape, made an enormous contribution to the IPv4 Internet with the first viable web browser, first viable server, something called SSL, and many other innovations.

We intend to make the same level of contributions to the IPv6 Internet with our global identity registry, IRP, the SixChat protocols, and Direct End2End connectivity.

Basically, this is a completely distributed PKI, which has not been done previously. It's a completely comprehensive PKI protocol that allows the applications to actually not just check revocation, like with OCSP, but to do all PKI functionality. This was what was missing in the PKI that we created at VeriSign that made it inadequate for client certs.

Again, like I said, with server certs, there's one per secure server. One PKI can actually handle that at VeriSign or any other CA for all the secure servers in the world. But with client certs, there might be a 1,000 or a million client certs per secure server. If we're talking about a secure server for a bank, they might have millions of clients. Each one of those needs a distinct client server.

So the quantity or the volume of client certs is remarkably larger than the volume of the server certs. Trying to do a single PKI to issue client certs to everyone in the world would be like trying to make a DNS server that could do address resolution for every node. There's not enough bandwidth or CPU power to handle that. It has to be distributed.

DNS is distributed to about 20 million nodes, our servers around the world, and so it handles the load with no problem. But it's still kind of tied to largely static servers and publishing the addresses of nodes not of people. It doesn't have the right kind of security, really. It really doesn't handle mobile nodes.

So essentially, what I've done is create a new infrastructure that's a distributed PKI with integrated address registry that allows devices to register their current IPv6 address securely, allows anybody else to retrieve that so that they can connect directly to them, and so on.

In order to actually make my secure End2End chat program, I had to go out and create this infrastructure first. So we're now productizing this, cleaning it up, and making sure it will fully work. We're getting some good buy-in. We've already won two competitions as the startup of the year for Asia. We've won a grant from the Singapore government to actually pursue this, and we also have a VC that's very interested in investing in this.

So we're trying to create the necessary infrastructure to allow people to actually make use of all these global addresses and make a leap past client/server architecture and NAT and private addresses and things like that to being able to have devices connect directly to each other.

My phone, for instance, would be able to connect directly yours anywhere in the world. Right now you can only do within a LAN in IPv4 because of NAT. If I wanted to connect to your device, we would have to be in the same LAN or I would have to go through some intermediary server.

Well, the days for needs for those intermediary servers are over with IPv6, but we need this new infrastructure to make it work. Basically, this is what we're trying to create here. We'd welcome any feedback from people that are interested in what we're doing here and have suggestions for how to standardize this, and so on.

Currently, this is something that we're pursuing on our own. We would very much like to coordinate with other people and eventually have this become an IETF standard.

We think this is a viable thing to be a kind of successor to DNS that will actually be more appropriate for the IPv6 decentralized, very-mobile Internet.

Okay, so that's basically my presentation. If you would like to get in touch, my e-mail address is on the first slide there. It's lhughes@sixscape.com. Feel free to get in touch. I'll be glad to send you more details. We'll be probably doing beta testing in the not terribly distant future. We would love to involve any of you that would like to see how this works and possibly even start interacting with us on it.

Questions?

EBERHARD LISSE: Thank you very much. Unfortunately we don't really have IPv6 in Namibia yet. I would be very interested otherwise.

All right, we're running a little bit behind time, so without further ado, Nigel Roberts with Securing Small Registries.

NIGEL ROBERTS: I think I'm going to use this microphone. How we doing with the slides? How can we get the slides here?

LAWRENCE HUGHES: How can you get the slides?

UNIDENTIFIED MALE: They'll put them in for you.

LAWRENCE HUGHES: Okay. Ask them. They'll be glad to give you a copy of the slides. Or I've got cards here. Get in touch. I'll be glad to [inaudible].

NIGEL ROBERTS: I was talking about my slides.

LAWRENCE HUGHES: Oh. Sorry.

NIGEL ROBERTS: I'm Nigel Roberts, CEO, CTO, and Legal Director of a small registry. The reason I'm talking about those different hats will become apparent. We're ccTLD managers for two ccTLDs, and we are the registry operator under contract for a third.

What's the story today? Well, I'm going to tell you about how as a small registry operator we found two or three useful technical techniques which we think are worth sharing. Some of you will just go, "Yeah, we know all about that." Some of you might never have heard of them, so I'm interested in feedback. Including one feature of the Linux kernel that was introduced in 2.4 onwards that really does help us sleep better at night. So let's get started. Can I have the next slide?

Okay, so what are small registries? Next slide, please.

Right. Here we go. So there was a story of Little Red Riding Hood, who is a small registry operator, her Grandma, and the Big Bad Wolf. Actually, no, there's no Grandma. Next slide.

So, small registries. A handy definition we've found useful is one that is less than 20,000 domain names. That includes many ccTLDs. It includes us, developing world ccTLDs, but also I guess one or two of the new gTLDs who are what you might call non-aligned. Sorry, that wasn't meant to be an intentional joke.

EBERHARD LISSE: One or two hundred.

NIGEL ROBERTS: What I mean by non-aligned is a small new gTLD in my book is one that runs their own shop. If they have just maybe got three registrations and they're part of one of the big guys, they're a big registry. But there are one or two. There's at least one I know that runs CoCCA.

What about big registries? Well, they've got technical departments, CT/OIT departments, risk management divisions, [inaudible] and so on. Usually in small registries, those hats are combined. But small registries face similar threats to larger ones, and they might be less able to cope. Next slide, please.

Does anybody recognize this guy? Anybody at all? Met this guy or recognize him?

EBERHARD LISSE: No, but I can read on the screen.

NIGEL ROBERTS: Yeah. His name is Kevin Mitnick. Probably you're all too young to remember him, but he was the arch-bad guy/arch-hacker, and guess what? He's now a security consultant.

Small registries combine roles and have to punch above their weight. So sometimes it means that they have the opportunity to be agile. As a shiny example of this – I won't look 'round – but .na and their leading role in DNSSEC. But other registries might still be dwelling in 1999.

Small registries are not so much of a target for kudos-wise as managing to hack a larger registry, but let's face it: all registries are trophy targets.

Reported hacks – I use the word "reported" advisedly because I'm certain; in fact I know there are several more – include .tp in 1999, .ly in 2011, .ro in2012, .nl in 2013, and of course, in 2014, ICANN themselves and their information system for new gTLDs.

If you put a new machine on the Net and start tailing the logs, within a few minutes, the bad guys are knocking at the door. Next slide, please.

Small registries might want to get management access to your servers. You're going to leave port 22 open? Not such a great idea. Put it on a different port? Well, everybody knows. Obscurity is not security. Changing ports will hardly help. Does anybody know a hacker who doesn't know how to use Nmap?

Disable route over SSH? Well, that's kind of always a good idea, but it still just leaves you with a honeypot, just asking for brute force attacks, [DOS], and so on.

How are you going to respond to the threats? Shut the computer down? Put in in a lockbox? That's the standard definition of a secure server, but even then, I'm not sure.

Let's go into the handy hints. As a registry operator, we did three things that gives us major enhancement and helps us sleep at night. Monitoring: know when something is wrong. This is Nagios, and this our live monitoring. It's not too unfortunate that you can't read most of it. I have actually obscured the host names. But we've done custom plug-ins for the services we run, things like POP3, EPP, DNS, and so on. If anybody's interested in the Nagios plug-ins that we have actually developed, drop me a line.

We also do SMS alerting. If ever one of those green lights ever goes red, about four us in the registry have our mobile phones go bananas at all times of day or night.

I'm not going to dwell too much on the second point, which is to enhance resilience and recovery. It's sort of motherhood and apple pie. Not just regular backups, but we do point-in-time recovery, log shipping, and we do real-time streaming. Next slide – oh, you're ahead of me.

We have two alternative registry locations, one that's active and one that's currently being built. We can activate that with a switch of a low TTLC name. That's an example of when a host name in the DNS can be switched within under five minutes, and the registrars basically don't even notice, unless they're in the middle of a transaction. That uses standard Postgres features, and it's achieved without modifying any of the registry code itself.

Extra fortifications: man the battlements. That, by the way, is where I live. Not in that building, but about a half mile away. Custom security perimeter: we all know what security perimeter is. IP restrictions: the application level restrictions. Those are default inside, for example, the CoCCA code and were not good enough for us.

Most importantly, they don't prevent the annoying doorknob rattle and brute force attacks and the resource problems that come with them.

So what to do? Well, as everybody who has used IPTABLES knows, firewall rules and managing them can be cumbersome. Port knocking: that's also cumbersome. They have their place.

But there are these things called IP Sets, which can be used in conjunction with firewall rules, make this easy and elegant. How many people here – and I'm sure there's going to be one or two or quite a few – have heard of IP Sets? Not as many as I'd expected. Not as many as I'd expected. Actively use them? Same people.

Okay. Inside Debian-based distributions, apt-get install ipset will install the necessary requirements. It's a framework inside the kernel since 2.4, which you can store IP addresses, port sequences, or IP/MAC address pairs in a way that ensures lightning-fast matches.

That allowed us to create what we call our custom security perimeter. To access all the various servers that we have, there are trusted locations which have absolutely no restrictions, that's just as if they're on the network in a way that you'd expect.

Registrars? Well, we definitely trust them a little bit, but they can see ports 443 and port 700 only. They can do EPP transactions and they can

reach the HTTPS we interface. But to anybody else, the registry system just appears not to exist. It's not in the network.

That's actually the example of the IPTABLES rules that we use, and the --match-set switch is what we use. The first one creates the trusted, and the second one is matching registrars with 443 and 700.

The advantages of using IP Set-enabled IPTABLES rules is it allows the management of sources separate from the rules. Fit the rules, forget them, and manage the contents in a different way.

So the sources are authorized IP addresses mainly, as far as we're concerned. They can be single IP addresses or CIDR blocks. IP blocks can be paired with ports, as you've just seen, or even Mac addresses, although we haven't found a use for that.

We created a very, very simple management tool to which to manage the IP addresses because obviously registrars come, go, change addresses. I'm not even going to bother to explain how that works. It's just an editor.

This is cool. Next one, please. Dynamic IP Sets. That's the one. Back one.

The IP sets can be created dynamically, so you can actually put in an IPTABLES rule which creates – the IP set here is called Pilgrims, or Seekers After Truth – and this enables us to put throttles on our WHOIS server. So we have different ones that implement on the same WHOIS server, Domain Ability Checker, and WHOIS2, and we can take different actions. We haven't really developed that too much yet, but it certainly put a brake on a certain amount of WHOIS abuse that we'd been seeing.

Finally, concluding the story of the Big Bad Wolf, we've built a big fence. The wolf went away to figure out another way in, leaving Little Red Riding Hood never to forget…He'll be back.

Thank you.

EBERHARD LISSE: We have two minutes for questions, so let me ask if there is one from the floor.

UNIDENTIFIED FEMALE: [inaudible]

EBERHARD LISSE: Sorry, the mic is not on.

MERIKE KAEO: Hi. My name is Merike Kaeo, and I just wanted to make a comment. I am leading a work party in the SSAC on registrant protection and credential management because one of the things that we've found is that there have been a lot of compromises in the last few years that actually deal with some kind of credential compromise in the entire lifecycle.

One of the things that specifically we want to address is some of the issues with small registries and registrars and some of the issues that they have. So just a solicitation for, if you have any comments on this, to come and speak to myself or some other SSAC member during this week. Thank you.

EBERHARD LISSE: Can you say your name again? I didn't catch it.

MERIKE KAEO: It's Merike Kaeo.

EBERHARD LISSE: Okay. Thank you very much. We find for our registry we are undergoing regular credit card compliance checks. Not that this is exhaustive, but it helps already to start looking at these things. Our data center hoster, who already left the building, has IP sets in place, and this is why I know about it.

All right. That's about it then. I will then release you all to lunch with the exception of the ones who want to participate to learn more about Adobe Connect. Josh Baulch will come in front and do that from here.

We'll meet at 2:00, more or less sharpish if you can.

JOSH BAULCH: All right, for those of you who are in the room here, we'll get started here in just a moment. This will just be a basic overview of Adobe Connect and how to connect up, so we'll get started here in just a minute.

All right. I'm going to go ahead and get started for those of you guys who are interested who are still in the room. What we're showing on the screen here is just an Adobe Connect room. What I'm going to go through is just connecting up to Adobe Connect via remote participation

as a participant. So any questions on just on how to connect your audio, that type of a thing.

What I'm going to do here is we're just going to kind of walk though this, and this is going to be fairly informal, so if you guys have questions, just shout them out and we'll go from there.

What you can see here is I actually have an Adobe Connect room open within Adobe Connect, so you guys can kind of see what I'm doing. As a participant, when I first get into Adobe Connect – this is me just logged into a room – one of the things you actually want to do first is go ahead and go over to Meeting, and the only option in that is Audio Setup Wizard. If you set that up, you're just going to walk through the steps that it asks. So it's going to play sounds, and then when you're doing this, you actually want to hear the sounds come out of your computer.

One of the things that we really recommend that you do is actually when you're interacting with your computer – and this would be in place of if you're in a location where you are unable to connect via phone or if Skype doesn't work reliably before you – then you're able to just connect your audio via Adobe Connect, similar to how you do Skype.

In this case, one of the things that at ICANN we recommend that everybody do is to have headphones when you are interacting on Adobe Connect. The reason for that is because the audio cancellation on computers is not always the same. You've probably all been in a meeting where you hear lots of echo and you hear people's keyboards clicking – that type of thing. So one of the things that we recommend is

that you just have a headset, preferably one that has a microphone and earphones for that.

I'm just going to show you guys this again. You're just going to go up to Meeting here, and you're going to go to Audio Setup. You're just going to walk through that process. Once you have your headphones plugged in, you just go through, play the sound, make sure you can hear it so you can adjust your volume on that, and then we're going to go onto the next step. You have the selection here of different microphones. In my case, I have four different microphone options. If you plug in your headset on most devices, not all operating systems, you just pick the microphone you want. So if it's your headset, then it will show up as Headset.

In this case, I'm actually going to use my built-in microphone. What I'm going to do is I'm going to go head and hit Next. In this case, I can go ahead and hit Record. You can see here that it's metering me talking, so you can hear it talking.

This is probably the biggest hurdle that most people face: they're not hearing any audio. So you just want to make sure that you adjust your settings accordingly.

I'm going to go ahead and hit Stop, and we're going to in back through, and I can play this. What this is doing is it's playing it back. I have my computer muted right now because otherwise it would be kind of obnoxious. In that case, we'll pretend that I heard my voice talking back, so we're good.

I'm going to go ahead and hit Next. You just go through this step that says Testing Silence. Basically what it's doing is making sure your audio is not too fuzzy, if you will, or have a lot of ground noise.

So I'm going to go ahead and hit Next. It's going to have it basically set up my Adobe Connect room for my audio, and then hit Finish.

At this point, you'll not see a microphone at the top. All you see here is a speaker. I'm actually going to make this big for you so you guys can actually see what I'm talking about here. There we go. In this case there is no microphone. What that means is that the ICANN staff or whoever is running – the Chairman of the meeting – hasn't actually started that audio for that conference. So what I'm going to go ahead and do is I'm going to actually bump over and start my audio for this.

So this is not a part that you guys would have to do. This is as a host. I'm going to go ahead and start the meeting audio. I'm going to bump back over to my other screen here.

Back as a participant – one more thing. One of the things that you may run into is that in some cases staff may not have enabled microphones for the participants. All you have do is just in chat just ask them to enable the microphones for you. I'm going to go ahead and do that now.

What you'll notice is in the attendee pod over here on the left – the right for you guys – you'll see a little microphone button here. What that means is that I now have the availability to connect my audio. You'll also notice at the top that the microphone is now available. I'm going to go ahead and click on it, and say Connect My Audio.

Now, there's another option down here too where, again, if you want to change the microphone if you have multiple microphones on your computer, then you can go ahead and select which one.

In this case, I'm going to go ahead and Connect My Audio. The way that you'll be able to tell if your audio is streaming in Adobe Connect is that if you watch up here on the microphone you'll see that there's little sound waves coming out of the microphone. That's how I know that my audio is going into Adobe Connect, so I know that everybody can now here me.

So you don't always have to ask, "Can everybody hear me?" because you will know based on the sound waves coming from that. And you can also tell down here, and you'll notice that the name at the top of the attendees pod there says ICANNRP25, which is who I'm logged in as.

In that case, that is pretty much the basics of connecting your audio.

Just some other things about Adobe Connect. If you want, under the chat pod, if the text is too small for you, you can go ahead and select and change the text size. I can make it so it's much bigger. I can also change my color. So if I want my comments to stand out from others, I can change it to red. Then if I type a comment here, now when it comes up, it's actually going to show up as red. It's kind of nice if you want to differentiate yourself in the chat pod.

That's pretty much the basics of Adobe Connect and just connecting. If any of you guys have any questions, I'll be up here. You can swing by, stop and talk to me here.

We also have training videos, a little bit more advanced training, on Learn.ICANN.org. It's just called Adobe Connect for Participants. It doesn't require a log in, so when you get to that screen, you can just skip that.

I'll go ahead and show you that really quick. Okay, so we're going to go there. What we're going to do is Search Courses, and I'm going to say Adobe Connect. Now here's the Participant's Guide to Adobe Connect. We're going to go ahead and just click on that link. It's going to ask me to log in. You can actually just log in as a guest. There's no reason that you need to actually log in and have credentials.

Now I'm here. Now there's actually video here. It's about a 13-minute video. There's also PDF instructions for you that walk through basically what I talked about, just step-by-step. Again, that's just on Learn.ICANN.org.

That pretty much concludes it. This was pretty brief, just kind of a quick rundown. I really appreciate your guys' time. If you have any questions, just come up here and talk to me.

Thank you.

EBERHARD LISSE: Marc Blanchet, where are you? There you are.

Okay, welcome, those who made it back. The other ones will trickle in more or less, I hope. In any case Marc Blanchet will give us the usual IETF update, and he's going to tell me now what RDAP means.

MARC BLANCHET: Good afternoon. I hope my talk will not have any issue on your digesting of the lunch. This is about RDAP. Let's see how it goes.

RDAP is a new protocol designed by the IETF "roughly requested by ICANN through EWP." RDAP means Registration Data Access Protocol, a replacement of WHOIS.

A few key features of it: it's structured data – yeah! – using JSON as a format, a modern query method using HTTP RESTFUL, flexible and modern data structure – for example, it supports internationalization – as a AAA model, identification, authorization, and accounting. Given that it's over HTTP, we have all those resources available. It's combined numbers and names access protocols, so it's the same protocol for both kinds of objects. I'll just give you a good post of Scott on CircleID recently that describes RDAP.

This is the incomplete RDAP tutorial in three slides. I'm assuming that you know what WHOIS is, so it's straightforward by examples since this is a tech session.

Numbers: you could request for example the first example up here is requesting an IP address. First line is requesting the information – the "WHOIS" information – for a /24 IPv4 address. Second one is an IPv6 address. Third one is an autonomous system number.

Names: as you can see, after there's an IP then autnum and then domain and then server. This is the actual special reserved keyword for those different objects. Then you put the object name after the /. So these are three examples of domain names, including IDNs.

You could query the information of a name server or an entity such as an handle in a registry. You could simple search. There were some debates within the Weirds Working Group, which is the IETF working group which worked on RDAP (because we're done), about having all kinds of search patterns, including Regex and all that stuff. We decided to split into very simple search patterns in the standard base protocol or specification, and then everything else being punted to another document that was never finished. So essentially we're back to basic search.

Here's an example of a response. I hope that you could see it on your own screen. I didn't want to go into the syntax of it because it would be pretty boring, but this is just a simple example of one answer. With pretty printing, it means almost 400 lines of JSON, or 8K of data. So it's very verbose.

But I think the best way to look at this is functionally what the response data returns, for example, handles; objects; details, such as numbers, names, name servers and stuff. Links – think of this as hrefs, which is essentially links to external URLs somewhere. Notices and remarks, so terms of use. You don't have access to that data, server is down for maintenance, or whatever. Events, such as when that object was created, last changed, and things like that.

Entities, an interesting part here is that we're reusing vcard format, which is also an IETF standard. Vcard4 was designed using the same traditional format, and there were two additional alternative formats for vcards: one was XML and JSON was made just in time or for RDAP. Whenever you have an entity that looks like an organization or a person

with an address and things like that, then it's encoded as a vcard in JSON, so it's a JSON-ized vcard within the big RDAP JSON response. Status of that object, locked and those kinds of things, and DNSSEC information and whatever else – all those things that we currently have WHOIS and more, but in a structured way so you can parse it.

Then the question is, "Okay, but where do I send the query to? Where do I find the server that could answer my WHOIS request?" This is known as the bootstrap problem, which is how to find the authoritative RDAP server for this object, which is reliably (from the right sources) and dynamically which whenever a new TLD or an address prefix is allocated, then you can find it. You don't have to wait for a software update or something. Flexible means that you could allow various services, such as how ICANN serves this as an RDAP server, both in HTTPS or HTTP or both or things like that.

During the working group, there were a lot of discussions about how we should implement bootstrap. There were different methods that were looked at. Two were done which were looked at as I would call in-DNS, which is either a resource record at the apex of the zone, or in a separate tree, for example, under the [inaudible]. There was also an IANA registry.

We evaluated those solutions based on various criteria, such as the capability to be used in Javascript, which is a pretty constrained environment. We need a new DNS resource record just for this, and it should be "simple." For example, we don't want to parse Regex expressions.

During that discussion, we hadn't found a method that was perfect, whatever perfect means, so we chose the least pain, hopefully.

The way to do bootstrap is that we agreed to have an IANA registry, and that registry will be populated based on the current IANA mechanisms for current assignees of objects, so it's modeled on how WHOIS entries are updated. When I say "entry," I don't mean the specific object entry, but I'll show you in a minute.

That registry will be in JSON format, so this will be the first IANA registry in JSON. Given this design, it will be pretty flexible for anything we want to do now or in the future. It's kind of straightforward and simple.

This is, for example, the IPv4 current address space registry, and as you can see, there is a column that shows the WHOIS server that responds for this address space. It's the same for IPv6. In the root zone database, there's also an entry for this. So the idea here is we would just add new columns or new entries for RDAP server and then generate programmatically the JSON registry.

So the envisioned process is that we would add a new RDAP entry into existing registries, the one that I just showed, and through current mechanism, to update TLD and address prefix records, IANA will receive the RDAP info and publish it in the new entries. Then the actual JSON registry will be separate, but will be refreshed programmatically with the new changes, roughly in real-time.

So think about those registries having a new column, which is: here is the RDAP server URL for these prefixes. It's the same for v6, and the same for TLDs. And obviously being on ccNSO a consistency here if you

have an RDAP server running, then you would send that information to IANA for updating your records, and then it will automatically appear in the JSON registry.

This is the example of the JSON registry, the boostrap RDAP JSON registry. This is the example for names. Again, I'm not sure I want to go into the detail of the syntax, but roughly it's [ARES of ARES of ARES] of objects, of strings, in JSON. It will have a version in which, if later one we actually change the format of the JSON registry, we would have a way to signal it, publication date and time, so you'd know when it was refreshed the last time.

For example, here's an example of .net and .com registries, who are together. The RDAP is the same server for both TLDs. Here's an ordered list of the RDAP URLs where you would append the suffix of the object you are looking for. If you remember previously with my first examples, if I want to query example.net, I would add to this URL "domain/example.net." So this is kind of the base prefix to append these specific RDAP keywords. This below is an example of an IDN-TLD.

It's the same for numbers. Instead, you will put the prefixes in, again, as strings in [ARES]. That's kind of straightforward.

Obviously, we can use redirect to another URL using standard HTTP redirect methods. A registry can point to another one. For example, in numbers there are places where there's some allocations in other registries, so they can refer back and forth between themselves using the redirect method. Obviously, this is a good thing of being on a modern access protocol: using the facilities on the "transport" protocol HTTP.

Just a few slides on non-IETF activities that are relevant to RDAP, which I just discussed. There were Interop sessions that were conducted during multiple IETF meetings on the Sunday morning. I think we did like about ten Interop sessions, if I remember well. It's been for some time.

We had about ten different implementations, and over time we saw a very good level of conformance to the protocol spec because we were using a comprehensive spec conformance suite that acts as a client testing against the RDAP server. So we were really testing the RDAP server's conformance as a client in the test suite. There were about 150 tests. Recently, we added a web interface so the people can run the tests by themselves using the web interface.

This is an example of an output of a test suite. You get the test number or string that identifies the test on the first left column. Then if the test passes, you get the actual URL, the actual HTTP request that was done, what the return the code, and the detail of the response.

When the test succeeded with a warning, then you get the warning, and if you click on any test, you get all the details of what happened, what was sent, and what was received.

We got pretty good comments on the usefulness of the test suite. It actually helped find the bugs and specs and various inconsistencies and improved clarity.

The test suite is still available for an implementation, so if anyone wants to get access to the test suite, please contact me.

That slide said there's also an RDAP workshop being organized for IETF in Prague this summer. This is bad because there were URLs on it, but

you could probably get it from the PDF on the website. There's a workshop being organized for Sunday before the IETF in Prague, discussing further down RDAP servers, implementations and stuff.

Conclusion: RDAP is a modern replacement of WHOIS, finally, using HTTP RESTFUL and JSON. Bootstrap is using a JSON-formatted IANA registry. We did Interop testing. A workshop is coming. Now let's implement it and use it.

So hopefully I'll be convincing you that this is great and you should implement it on the ccTLD side of domain names.

These are the references of the drafts that are currently in the RFC publication queue. At any time, they may become RFCs. They are considered as a cluster, so they may appear all at the same time for RFC publication.

That's my talk. Any questions?

EBERHARD LISSE: Thank you very much. Any questions? I see Jay coming. In the meantime, the presentation will be uploaded. That page missing, we will ask you to just send it to us again so that on the uploaded presentation, the links will all be there.

MARC BLANCHET: It's already there.

KRISTINA NORDSTROM: It works in Adobe. It's working. It's already there.

EBERHARD LISSE: Okay.

MARC BLANCHET: Oh, okay. So you saw it?

KRISTINA NORDSTROM: Yes. I don't know why it's [inaudible].

MARC BLANCHET: Okay, whatever. Thanks.

JAY DALEY: Jay Daley from .nz. For the bootstrapping, because it's specifically designed to work within Javascript, I presume that that means that potentially client software will be following the full chain, from the bootstrap down to the new RDAP server.

MARC BLANCHET: Mm-hmm.

JAY DALEY: Has any work been done to estimate the volume of lookups that will be done on the RDAP servers, and has anybody considered whether it is useful for us to put ICANN into the operational chain once again in a new way?

MARC BLANCHET: Good point. Before answering fully your question, the in-DNS solutions were great for that. It was just a no-brainer for one of those features. This was just done. IANA registry had this issue.

During the whole process of this, we had a close relationship with IANA technical people about that, and they already have CDNs for some of those registries that are hit very often, kind of a large volume. So that problem is "solved" in a sense that they will be using CDNs as they do right now.

JAY DALEY: Do we know what the large volume of the other registries is and how that compares to these large volumes.

MARC BLANCHET: Sorry?

JAY DALEY: Do we know what the large volumes of those other registries are and how that compares to the large volumes of this?

MARC BLANCHET: Expected? I don't have those numbers, but this is CDN stuff, right, so it scales as you need. It's a no-brainer. So you're not going to it, IANA serves, you're going to it – I don't know, [Akamai] or whatever – and I don't want to say…

JAY DALEY: No, no, no. I'm not worried about the technology. I know the technology works. I'm worried about the structural impact on the Internet of having that many queries now that important handled by one single organization again that has an expanding role in that area. Now, that's not necessarily for you, but I'm just wondering if the group had done any analysis of what the size of that would be.

MARC BLANCHET: No, we haven't done any specific volume statistics or expected statistics.

JAY DALEY: So 100 billion lookups a day is not unreasonable then?

MARC BLANCHET: What I've been told from a few people is that there's no difference here than an HTML file located in CDNs that could be a front page of a very well-known website.

There was a concern in a sense that people were raising the same question you had, and the answer was, "Okay, if IANA has a contract with CDN providers, then what's the deal?"

JAY DALEY: Because we're all going to have to pay for it. That's the problem.

MARC BLANCHET: Okay. Thank you.

EBERHARD LISSE: Any other questions from the floor? All right, Marc, thank you very much.

MARC BLANCHET: Thank you. Hopefully you will be on RDAP soon.

EBERHARD LISSE: Well, if it is easy to implement software, we'll make a plan. Okay, Vicky Risk is the next, from ISC. She will tell us about the decommissioning of the DLV.

VICKY RISK: Hi, everybody. I'm Vicky Risk from ISC. I just want to talk about whether or not is time perhaps to sunset the DLV. I've never been to Singapore before, but when I started doing searches on sunset pictures in Singapore I got a very good impression of this place.

Oops, I'm doing it wrong already. There we go.

I'm just going to really briefly explain what the DLV is. I apologize if I'm boring those of you who already know what it is. The DLV was set up initially as a transition mechanism to aid in DNSSEC adoption, pretty far along in the DNSSEC adoption curve. You can argue about that, but I think as far as the original of the DLV, it's probably served its purpose and it is probably time to plan for gradually taking that out of service.

I'm just going to briefly outline. It's a little bit complicated how we actually take this thing down because we've got a lot of systems in place now supporting it. I'll discuss the timeline that we're proposing.

The DLV was originally set up really to address a chicken and egg problem, which was that without the top level domains and the root signed, you could sign your zone but nobody could validate it, so we needed a way to allow people to start DNSSEC signing kind of at the leaves of the tree.

I'm not going to read this. I just put that in for people looking at the slides. Whoops. There we go. It's censorship again. I saw it happen to the last guy. It's censorship. But I know what that picture was. It was a picture of the way that DNSSEC is supposed to validate up to the root, and the point is that prior to the root being signed and prior to the TLDs being signed, this was the substitute path to validate your DNSSEC-signed zone.

I see this is going to be a test.

So the question is, is DLV now possibly actually impeding the continued deployment of DNSSEC. The benefits of the DLV are that it allows a signed zone to be validated even if the parent is not signed, and it will accept DS records from anyone. We have a little web portal. Anybody can do it.

The disadvantages are that of course it reduces the pressure. If you want to sign your zone, you're not as motivated to pressure your parent to sign if they're not signed, and it reduces the pressure on registrars to accept DS records.

The other significant disadvantages: all of the resolvers that are validating out there that are querying the DLV, that's an additional query that they have to do.

Who might mind if we decommission the DLV? Who needs it? Entities with signed zones under unsigned parent zones, and entities whose registrars don't support DNSSEC.

This last use case is something I'm actually looking for feedback on. I honestly don't really understand it, but just recently I got an actual phone call from somebody who called ISC who is trying to use the DLV, and he needed to put his records in the DLV for two weeks while he transitioned from one registrar to another. I hope that this isn't a really big problem because if it is, I'm not sure when this problem goes away. But if anybody wants to talk to me about this offline, I'd be glad to talk about it.

Since this is a ccNSO Tech Day, as you all know, the DNSSEC adoption among the ccTLDs is really very good. I couldn't help putting in this picture. This is from the Internet Society's Deploy360 site. They have maps for every geography.

This is less visually exciting. This from ICANN. This is as of October. So not only were 76% of the TLDs signed, but also the angle of that line is really nice. It's pretty steeply up to the right.

I was just surfing around on the ICANN website. This apparently was just posted. I'm not sure who looked this up, but I was happy to steal the data. This is a little bit more recent than the prior slide. That's why the numbers of TLDs are different. But as you can read, of the 816 TLDs in

the root zone, it said that 636 TLDs already have trust anchors published in the root. According to ICANN, five TLDs have their trust anchors in the DLV, and none of those five need it.

The second big issue besides whether or not your parent is signed, of course, is whether or not your registrar will accept DS records. I did find that ICANN has a page where they keep a list of registrars that will accept DS records. It looks to me like it's probably not been updated very well. It seems to me like it would be useful if people would want to help fill that out a little bit more. They provide an e-mail address for providing information if you care to give them a longer list of registrars that will work with DNSSEC. It's a pretty short list up there.

So are we ready to sunset the DLV? The root signed. A large majority of the TLDs are signed and have trust anchors. There are registrars that support DNSSEC, although they don't all. It seems as if to the extent that there are registrars that won't accept DS records, announcing that we're sunsetting the DLV would help to put pressure on them, so it'd be a good thing.

What are the things that we have to do to turn off the DLV? We have three basic elements. We have the little registry that we run and the records in there, which belong to people. We have the resolvers out there that are validating and the operators of those validating resolvers, and we have the zone itself.

The first thing we have to do is empty out the zones that are in the DLV. These numbers are actually about six months out of date, but a surprising number of people who try to use the DLV somehow don't

manage to complete it successfully. But we have about 2,800 working delegations in the DLV. Only 397 of them have an unsigned parent.

Of course, the first thing we like to do is notify all of those people that in the DLV that we think don't really need to be and ask them to go ahead and remove their records. We want to stop allowing people to add new zones to the DLV if their parent is signed, and eventually we want to, after we've given ample notice, remove the records.

The timeline that we're proposing starts out with notifying people who we think could remove their zones and give them some time to remove them. We'll probably send them repeated reminders. Sometime in the middle of this year, we would like to change our interface for the DLV and no longer allow you to register a new zone if it could validate otherwise.

I got a little thing on the slide here. Then we would like to purge those zones sometime in 2016. So we should have been able to give people at least a year's notice. That should be enough. Then sometime in 2017, we'd remove the remaining DLV records.

I mentioned one of the downsides of the DLV is that it requires that validators perform an additional query. We can't stop doing that in the resolvers until we have removed the records from the DLV. Otherwise, of course, you'd have people who would have validation failures. We do recognize that there's a very long transition time if you want people to stop using a feature that's supported in their software that they may have deployed and forgotten about a long time ago, so this is going to take a while.

There also are many, many, DNSSEC How-To guides out there to tell you to use the DLV, and in some of the distributions of the software, it's enabled by default. So these are some of the things that we'll need to unwind.

This is just showing you in the Redhat, the CentOS, NetBSD and Fedora distributions for both BIND and Unbound that DLV is enabled by default.

The zone itself, we'll have to keep up until we can see that nobody's querying anymore because it would cause some adverse impacts, let's just say, to remove the zone while people are still querying it.

The plan is to go around this year and tell everybody we want to do this. We'll have to write an application to start notifying people whose records we think don't need to be in the DLV. We'll probably have to provide people with some information, some how-to, about migrating. We'll have to find some of the popular DNSSEC How-To guides and try to get them adjusted.

Eventually – this is longer term – we'll have to try to remove the default enabling of the DLV in distributions with resolver operators.

I don't know what was here. Anyway, more censorship.

That's it. That's all I have – oh, I see Dan is already getting up.

EBERHARD LISSE: Dan?

DAN YORK: It's Dan York, Internet Society. What you said about talking to the – we were pointing over here at Paul but just because I think there are folks in this room who can help you with turning off some of that by default, we hope, or at least file bugs to move that along. Well, I don't know. Okay, it's a larger question, whether we want to.

But back to your question about registrars, that list that ICANN publishes has been relatively up-to-date. The folks from there have been trying to keep that up-to-date at different times. One of the challenges that we've seen is that there are a good number of registrars who, even with the 2013 RAA and the operational requirements there to support DNS records and so on, still don't provide an easy mechanism for that to happen, so there is still this registrar stumbling block that is out there that we're hoping, as more customers request, more people want that. It's one of those challenge that we'll see.

VICKY RISK: But do you agree though that if we announce that we are going to decommission the DLV, it will only help? Does keeping the DLV up, do you think it's a reason to keep it up?

DAN YORK: No. I would prefer DLV to die, personally, so I'm all fully in support of it. I agree that it will help in some areas with pressure on that, but I don't know that the registrars really care. That's been my experience, quite honestly, in interacting with a good number of registrars around that. They have enough other challenges on their minds that DNSSEC is on

their list somewhere, maybe. So I don't know if this will help, but I think in other avenues it will certainly help.

VICKY RISK: Is it reasonable to ask with a show of hands if there's anybody who thinks it would be a mistake to sunset the DLV? Okay, so that's a zero. All right.

EBERHARD LISSE: Wow. Unanimous consent.

DAN YORK: I see others here who are wanting to chime in, so I'll [let it go].

VICKY RISK: Okay.

DAN YORK: We did have a discussion in one of the recent times about using the DLV as a mechanism for some of the workarounds that people were looking at. I think a lot of those are now, "Sorry, if your registrar doesn't support it, you've got to figure out how to go and use a different registrar or something for the TLDs that are signed at a top level."

But I know that there have been some people looking at it. I understand the value it can provide in a test bed environment, but I think there are larger issues with keeping it around.

VICKY RISK: Somebody get some phone books for Warren.

WARREN KUMARI: Somewhat of a loaded question: what happens if you just turn off the DLV servers and go home?

VICKY RISK: Some bad things happen, yeah. We're not going to just turn them off.

EBERHARD LISSE: [Paolo's] first.

GEOFF HUSTON: Geoff Huston, APNIC. We've done an awful lot of work about handing lots of people deliberately broken stuff that invokes serve fail and looking at the additional issues that come along.

The first news, which is actually really good news, is that one half of the clients who get a serve fail then go to the second resolver that doesn't do validation to go and get the answer. So if you turned it off tomorrow, the impact isn't as bad as you think because one half of the folk who might have been impacted simply say, "Well, I got the second resolver, so I'm not going to do anything."

That other half, that other 10%, will not resolve the name because of serve fail because they can't find a validation path. That's certainly true.

The next kind of question is: how long does it take, and how many additional queries. It takes another four queries, on average. It takes

about two seconds. So on the whole, it's not that bad. Yes, it affects around 10% of folk, but I certainly think in some ways it was a hack to get it started. We've started. It's now a case that registrars and everyone else need to get with the story. It's time that ISC stops spending its valuable money on an operational resource that is really just a Band-Aid around other people's laziness.

VICKY RISK: I can see that, but in fairness, it's probably none of us in here that are using the DLV.

UNIDENTIFIED MALE: I think that there's another issue of one where to just turn it off and not go through the correct sunsetting period, which is a lot of recursives who go and query it would suddenly end up with really large query queues and fall over dead because every query there now has outstanding one for as well.

VICKY RISK: Yes, there's some problems with just turning it off.

PAUL WOUTERS: Paul Wouters, Redhat. I am personally responsible for half your list, so you're welcome. But actually, it helped us kickstart things and moving DNSSEC forward.

I'm kind of sad to see it go because there are still so many companies that are too big to change registrars, and there are departments who

can't change it, and for those kinds of situations, which are extremely common, the DLV was very useful.

But on the other hand, I agree that we should kill it. I'm a little worried, actually, about your two years because if I look at operating system updates we do, we better get those updates in fast now so that in two years from now that's still working because some versions for instance of older, let's say, Fedora or Redhat releases will not be scheduled for updates.

VICKY RISK: So the plan is to remove the records in two years. My expectation is that it's going to take a while to get the installed base of resolvers to stop querying. But we can, of course, see if we're getting queries.

PAUL WOUTERS: Right. Because for instance if you're looking at, I don't know, [Redhat] version X5, it might not even see a BIND update anymore. So the configuration will also not change.

Plus, even if we update the OS, the problem is that if they've made a single change in their BIND or Unbound configuration file, the update will not actually override the config file, so we will actually have to specifically go in and see if we and disable that and basically rewrite their config files for the DLV part.

VICKY RISK: Yeah, that's why I said there's a real long tail on the resolver install base because I'm aware that, especially your users, love their ancient software.

PAUL WOUTERS: Right. It's very stable.

VICKY RISK: Stable. Sorry. I think that's it.

EBERHARD LISSE: Okay. Thank you very much. One shouldn't applaud, if a functioning system that has helped us all very well, too much.

All right, Mon Perez from .sg. Hello. How are you?

MON PEREZ: Hello. Okay, sorry for that. Good afternoon, everyone. I am Mon from SGNIC, and I'll be presenting not plainly on the business side of business intelligence, but mainly on how we implemented BI in SGNIC.

Okay, the next screen. Okay. This is just to show you what are the contents of my slides. I'll be presenting about the background on why we implemented BI, and then some of the benefits that we were able to get from it, and the lessons that we learned while we were implementing it, and what is next for BI to us.

Okay. The background: why did SGNIC want BI? Basically, I guess it's what every organization's purpose is. It's to gain some insights from the registration data.

What are those kinds of insights? For example, domain registration trends or profiling registrants, meaning SGNIC wanted to know whether there's a trend in how people buy domain names. Is it mostly government buying domain names for band protection, or is mostly wholesale or retail, or is mostly just anyone wants to get a domain name?

After the management approved the project, then the typical project lifecycle kicked in. We basically gathered the users' BI report requirements, and then we conducted a PoC with several BI vendors. We met with several BI vendors and then did an NDA to give our data to them and then have them massage our data and show us some of their BI tools and the capabilities.

Then after the PoC, we then threw in a vendor. We chose the vendor based on the one with the most intuitive solution. The intuitive solution is the key requirement that we had for this project because we believe that BI isn't for the technical guys, but it's mostly for the business users, so there won't be any use if the BI isn't that easy to use for the business users.

After we have chosen the software, then the project deployment commenced. We extracted and analyzed the data from our registration system, and then we created the reports and then published it to the end users.

Here is one of the reports that we have. Note that the values here are bogus due to sensitivity. This is one of the reports we have. Each row is a registrar and each column is a time field. Each cell is the number of registrations for that registrar.

The benefit that we got from this report is that basically at a glance we know right away if any of our registrars are performing good or if any of them are having problems based on the shade of the color. The lighter the shade is, the lesser the registration. The darker the color is, the more registrations they have.

For example, let's say for Registrar N there's quite a month wherein the registrar didn't do any registration, so maybe from there the marketing person can talk to the registrars, and then maybe see if they need some help.

Another benefit is more on detecting possible abuses. This report shows the registration trend for a certain promotion. The promotion ran for a month. A couple of days before the end of the promotion, there was a spike in the registration. Because of this report, our admin staff were able to see the spike, and then from there, we were able to do some investigations on whether it may be used for abuses.

So we checked with the registrars and did the necessary things. We asked the registrar to check with the registrant on how he intends to use these domain names. Luckily for our case, it turns out that the registrant was doing it for brand protection, so that explains why he registered more domains.

Not all of the reports that we do really make sense, but since we have the tool and we're able to visualize, as long as we have the data, then we just basically do it.

This is one of the reports that we did. It shoes the distribution of the registrant by country. Obviously, we know that the majority of the distribution will come from SG, but we still wanted to visualize it. Why? Because maybe from the marketing side or from the end users they can see some potentials in the other countries, and then maybe from there, do some marketing or do their stuff.

This is one of the interesting things that we've done when we went into BI. The original requirement from the company was to be able to identify the type of business that the website is doing based on this content. For example, Domain ABC, classify it as a, let's say, wholesaler or a retail-type of website.

While doing the PoC, we've talked to some vendors on whether they can do this stuff. Obviously they can do it. It's just that it costs much. So we did some research on how else we could do it. We did it through sampling.

After doing the research, there was a sampling calculator in the Internet. You just have to specify these three parameters – the population, the confidence level, and the error rates – and then it will give you a sample of domain names that you can use to at least get this kind of report.

This was one of the interesting things that we did. After getting the sampling, we actually hired a temp staff to do the manual classification.

Imagine a temp staff visiting 15,000 domains to manually classify it. Then from there the data was put into the database and then we extracted this report. This was the report as of August 2014.

The benefit for this is it can help the marketing do more targeted campaigns. Let's say based on this majority from the wholesale and retail, our marketing can craft some promotions that are targeted at wholesale and retail trades.

Another benefit that we gain from BI is there's now a clearer scope or division of work for parties in the BI workflow. Traditionally, most BI fails because IT knows that IT does the report and then end users keep on changing and the bottleneck goes to IT. So that's usually the pitfall of BI.

In this case, we knew about the pitfall, so we tried to educate our business users that, "Hey, this is the BI system. This is our scope. For the IT, we just generate the data for you guys, for the business users, the tool is intuitive enough, so do your own analysis."

These are the lessons that we have learned. Firstly, you need to get support from the end users for your projects. Otherwise they will just think that the project would just be another IT toy.

Second is you need to understand your data. No matter how advanced the BI system or any of your systems are, if you don't understand your data, then you might end up just creating reports with no substantial meaning.

Third is to recognize the differences between BI and BA. Before we ran into the BI and the BA, we really didn't know the difference. But after

talking to some of the vendors, most vendors usually sell you BA because they will earn more if you buy the BA side of their solution.

So I suggest that if you just want to churn out reports, just want to visualize it, stick to your BI plan, and then if you think that you want to move into BA in the future, then maybe buy a BI software that has the capability to expand to BA.

Lastly, do research before you buy the system.

What is next for SGNIC? Since we already have the system and we are already analyzing internal data, we're thinking of integrating more data or maybe buying data from external parties. Examples are maybe let's say a website ranking. Then we'd correlate it with our domains.

Another thing is, with the hype in big data, we're thinking of merging operational intelligence or those data from web server logs into our BI system. Luckily, the BI system that we purchased can actually support unstructured data.

I won't be able to do a demo, but I will just show you a screenshot – sorry – because I realize if I show you the system, some of the data are sensitive.

Let me go here. Okay. Oops. Okay. This is actually the system that we purchased. I think for some it's quite common. When you search about BI, this vendor was a leader in Gartner's Magic Quadrant. That was one of the key differentiators in other BI vendors.

This is the system that we are using: Tableau. On the left-hand side, you can see the database fields that we have prepared for the business

users. Below is called The Measures, but for some that are new to BI, measures are simply just accounts; for example, account of the domain names.

This is the typical view for the end users. From here, they can just simply drag the measure that they want to know; for example, they want to know the domain registration over time, so they can just drag the domain registration and then just drag also the time. If they do that, then it will look like – what's the next here? Hit next. Oops. Sorry. I'll just go back.

So after dragging, this will come out. This is what happens. Simply it will just show the number of registrations over time. Actually for this one, I added an additional parameter to split it by category or by extensions.

Other than visualizing the data, this software can also perform quick calculations which may be useful for end users. For example, the end user wants to – sorry, that's not the one.

For example, the end user wants to see the percentage difference of the registration per month, so they can just right click on the measure, or the domain registration in this case, and then select Quick Table Calculation and end up with the percentage difference. The percentage difference will show up in the report.

For example, for this plot, the percentage difference to the next month is about -66.2%. So anyway, from our side, from the technical side, we really don't understand much of these figures, so it's more for the business users to use.

That's it for me. Any questions? I hope you get something out of what we did.

EBERHARD LISSE: Thank you very much. I have a question. Why do you just go [inaudible] and purchase some software?

MON PEREZ: Open source.

EBERHARD LISSE: Why don't you look at it yourself and then use some tools like Arrow or another statistical analysis to do it in-house? There must be a consideration that led you to do that. What was that?

MON PEREZ: Okay, first we tried to see if there was an open source solution out there. Actually, there is, but then we realized some of these open source solutions are really more catered to the technical guys, meaning if you want some advanced reports, the burden goes to the technical guys. We didn't want to do that. We want the analyzing of things to be done by the business users, so for the technical guys, we just give you the data, then the business users do the analysis. We don't the business users to say, "I don't like this graph. I don't like this presentation. Can you IT guys do something?"

EBERHARD LISSE: Thank you.

LUIS ESPINOZA: I have the same question on my mind. Why not try some other open source software? Do you evaluate Pentaho, for example?

MON PEREZ: Yeah, we did. We did.

LUIS ESPINOZA: Okay. Very user-friendly, is this?

MON PEREZ: Yeah, I did evaluate Pentaho, but I think when we ran it we saw some technical errors. We realized that this may not be good for the business users.

LUIS ESPINOZA: The other question is – maybe I misunderstood or maybe I'm wrong – I heard in your presentation that for [analyzing] logs, you will buy another model or something like that and so forth. Why not put the logs into SQL and analyze it with the same tool? Just a thought.

MON PEREZ: You mean for the other data sources like the logs?

LUIS ESPINOZA: Again, you mentioned in your presentation that in order to analyze logs for a web server you need to buy an extra model of the software to analyze the structure of that.

MON PEREZ: Actually, I'm not sure if I said it, but currently the system that we bought already can support analyzing of logs from our web servers, so we really don't need to buy an extra plug-in.

The buying of the extra plug-in thing that I mentioned was more on the analytics part. If you wanted to do some statistical analysis, the current BI system that we purchase can actually support it. So my suggestion is if you intend to buy a BI system, try to think of what's next when you try to think of the BA, because after you get the BI, then you want to get more out of it, so then you move to BA. Okay?

EBERHARD LISSE: Okay. Thank you very much. As I said before, due to popular demand from Andre, we're going to have a short coffee break. So about at half-past we meet again, and then Luis and I will dazzle you with some flashy cards.

MON PEREZ: Okay, thank you.

EBERHARD LISSE: Thank you.

Welcome back. We actually had some coffee. Can everybody who wants to listen sit down please? The last item on the agenda is basically how to put a key in hardware. Let me pull out that hardware. I usually carry it around. That's the hardware. It costs much less than an HSM. It's a little bit, but not much, but a little bit securer than having the key in software.

The reason for this presentation is that I have my key in software since I think 2007. We actually wanted to go the next step and see what we can do to put it in hardware, but we cannot or don't want to really just yet because there is very little demand in investing into these expensive HSM machines.

So what do we do? Fortunately, Luis and I basically did the same things at different places. He worked for NIC-CR for a while and did fancy stuff with TPMs, signing of records in TPM. That's a chip that's on a computer, not on my Netbook, so when I tried it, it didn't work.

Eventually he and his wife always wanted to come to Namibia, so this time we could organize that the flight was going from Costa Rica to Singapore through Namibia, so we had him there for a few days, and we had a look at it. We experienced quite a lot of fun. Luis will do mainly the presentation. We have some practical things. It went much faster than we thought it would, so we need a little bit less time than we booked for. But Paul Wouters is willing to move up his presentation by 20 minutes without much prompting. Okay?

LUIS ESPINOZA: Good afternoon. This is Luis Espinoza from .na – no, sorry. Well, I accepted an invitation from Eberhard to play a little bit with this. I always accept an invitation to play with [technical] staff, especially with these DNS-related things. With DNSSEC, it's better.

DNSSEC is easy. This could be a [meet].

UNIDENTIFIED MALE: First slide ever that has said that.

LUIS ESPINOZA: Yeah. No, I would show you why it's easy. Is it secure? Look from the perspective of [inaudible] signing. DNSSEC provided [inaudible] signing tool records in this manner. [inaudible] signing can bring [inaudible], non-repudiation, and the other characteristic is integrity of the information that is transferred to the [wire by putting] some way to say it. Then, yes, I think it adds security to DNS.

Secure DNSSEC is expensive. Well, some of you remember some of the presentation from the Brussels meeting. There's some high, high, high costs hiding in some of the implementations. But at the end, doing this presentation and demonstration, we will show you that maybe it's not that expensive. We think about DNSSEC use and so forth. Well, less expensive, too.

What are we looking for with this implementation or this exercise? An easy, secure, and cheap DNSSEC solution. By cheap, I mean low cost, yes, but related to something else is management of DNS or a ccTLD. Then from that perspective, yes, it could be cheap.

The target is .na for demonstration purposes, and for fun, truly. The parenthesis on line 19 will show you why it's for fun.

EBERHARD LISSE: Maybe I just wanted to make one or two comments on DNSSEC being easy. It is easy. A gynecologist can figure this out when he's six weeks on his sick bed. It's easy to figure out. It's easy to implement, but it's secure. It's easy, cheap resolution that's not available.

If you put the key in software, and keep the key in software on the computer, then you have to really make sure that the computer cannot be broken. But it's not possible to be 100%. They come up with stuff every week. It's easy to do, but not easy to do in manner that will resist auditing.

When your bank comes and says, "How do you do this?" you can say [inaudible], "No, that's not good enough. We want this, and this and this, and there are standards for this," and then it becomes expensive. Then it becomes difficult. It's not only the hardware that is expensive. It's also the whole auditing around it that's expensive.

I understand for example that with Open SSL, if you want to self-sign a certificate, it's easy. If you want to have it audited, it's $70,000 U.S. dollars. For us, that's $700,000 Namibian. It's about ten salaries of a nurse per year.

So DNSSEC is easy. It's just not if you want to do it in a secure manner. It's just not that cheap. But we can play with this, and that's what we did.

LUIS ESPINOZA: Okay. To put the things in perspective – I apologize if somebody gets bored here because this is very basic stuff – but in some way to explain why it's [not] so difficult. Usually, you have a registry system which has a database. From the database is generated the BIND-type tables. Then you use updates, the serial number, and load the zone.

What happens after that is according with each implementation. Many people have many other servers, secondary servers and things like that.

When you have DNSSEC, it's simple like this. After you [inaudible] the BIND-style tables, you sign the zone files, for example, the DNSSEC-signed zone. When you sign the zone file, you can choose to use software keys – that is very common – or hardware keys. Then after that, you update the serial and reload the zone.

It seems here it's just an additional step in the whole process. It's not a huge change. You're not changing everything. It's not a renew of the model of how the registry handles. It's just inserting an additional step in this. You can keep going with the rest of the process as it's doing.

Probably most of the ccTLDs that run by itself the DNS know this stuff very well, and each of them, or each of you, has figured out the best way to manage these steps. In those cases, it's very easy to figure out how to start the DNSSEC signing.

Some of the management of the TLDs probably hire a company to handle all these details and steps. And, yes, it could be expensive from the point of view of that company who will charge you an additional

cost to handle this process of signing. But from the technical perspective, it's just that.

Now, how many of you are running DNSSEC in your managements or domains? Yeah. From those, who many of you are doing it on hardware?

EBERHARD LISSE: And how many not in hardware?

LUIS ESPINOZA: Yeah. And how many of you are not doing it on hardware, just doing it on software? That does include me. I have a domain running with keys in software.

EBERHARD LISSE: Me too.

LUIS ESPINOZA: Yeah. From those who are running DNSSEC in software, who has less than 10,000 domains? Yeah, this is for us. This is for us because this process could be very easy to implement if you have less domains because if you have a lot of domains, you have a lot of signing to do and maybe this smartcard is not powered enough. It's something like from two to ten signs per second. If you need to sign a million records, it could be a problem. But if you need to sign 200 records, it could be worth it.

EBERHARD LISSE: It's not only the ones that are present. There are 54 African countries and a few territories of which two do zone signing themselves. So there are 52 who couldn't be bothered or who feel overwhelmed or afraid or who are waiting for donors to give them money so somebody else can do it for them. But it's important that even smaller ccTLDs or smaller gTLDs, the new gTLDs, especially smaller ones – I've said this 1000 times; for everything they have to do they have to give a Euro per domain name to an external provider – DNSSEC, escrow, registry, registrar, and whatnot.

This is not rocket science, and I'm always overwhelmed when I see that people who have got countries where there are many universities couldn't be bothered.

LUIS ESPINOZA: Okay. Well, let's talk a little bit about hardware signing, hardware keys. In the top right off this slide, we have the formal HSM, whose cost is something like from $3,000 or maybe a little bit more to $25,000 for only the box. This is only the box, and only the box is not the most expensive thing to implement the DNSSEC.

On the left side, that's what I like. It's a TPM signing. Until I know, .cr is still running, signing on online hardware keys using PPM. PPM is cheap. There is an ambit in many systems, including the servers. From the perspective of the cost of the hardware, this is the cheapest technology to signing hardware because it is an ambit in the server. It doesn't have an additional cost.

But buying the hardware is not the only cost. This is important to take into account. I think understanding very well the process is more important than the hardware by itself and not buying a box just with the solution. Yes, it's one thing to understand [inaudible].

Now, the thing we bring here is the model of the [right now] is the smartcards. With a smartcard reader, we can see some of them here. This is some model of a smartcard reader. This is funny because this is our two models. This is an antenna and that one is SC?

EBERHARD LISSE: SCM, and this is the latest model from Amsterdam.

LUIS ESPINOZA: Yeah, SCM is the brand of the smartcards. This one I went to my bank to buy a simple smartcard reader for the national signature. I didn't say what would be the use of this smartcard. They sell it to me. I tried it, and it works. The thing with the smartcard reader, only they read it. It's not a big issue, just to be very standard.

The other surprise we had was as soon as I plugged it into my MacBook, it did recognize automatically. Maybe after the presentation some of you can try. You plug it in and it really plug and play. You don't need to install drivers or install anything. The smartcard reader is just very easy.

We tried three different smartcard readers in this table. The idea is that this smartcard reader is very standard and cheap, so cheap as hardware. It's very cheap, really. The critical thing here is the smartcard by itself. Each smartcard has a different internal operating system.

For the purpose of this table, we are using OpenSC. This is open source software to manage the smartcard. We need to find the precise smartcard that works with OpenSC. This model of smartcard, from a German company, works very good as HSM. It's a better model of HSM in the smartcard with this software. It's a good thing to check the software and see which smartcards are running very well with the software.

For example, we tried to read a credit card and if the physical contacts match, you can see the smartcard trying to read the credit card. But you cannot read what is inside the credit card using OpenSC. It's a different protocol. It's a different operating system within the card.

This part is important. Although, the good news is both things you can by in units. You don't need to buy in the thousands or hundreds of units. You can buy them in smaller quantities. You can buy for play at the beginning. As we will show you later, you can have all the environment to try the signing and try the generation of the keys on your laptop. Everything is available.

Well, this is a little bit what we're talking about. There's another branch of markets that do work with OpenSC. For example, we tried some [VTN] brand. I keep it from a workshop a few years ago with Richard Lamb. With the smartcard readers, we tried different brands, and almost all of them work.

The software. This is very simple: OpenSC: OpenSC-project.org is the link of the website.

EBERHARD LISSE: Check the links. It's all in there.

LUIS ESPINOZA: Okay. I will check that right here. I will check. Yes, quickly.

EBERHARD LISSE: All lines in blues are hyperlinks .If you want to access it, you can just click on the link and it gives you the obvious website of the product they're talking about.

LUIS ESPINOZA: Okay. You can see here there's a link to download the source and the Windows version, and there's the Mac OS X installer. We use the Mac OS X installer. They have support for Yosemite, the latest version of Mac OS X. It's very easy.

Okay. Then we need BIND, obviously. You can use Homebrew or MacPort to install it. It's very easy. It works like [APT, GAP, or Jung] in CentOS. You just type the command, install the ring of the packets [inaudible] node, or you can use the sources of BIND. It could be BIND 9.9 or BIND 9.10. It doesn't matter.

We need to do a little trick here. We installed [inaudible] with an image of CentOS that has all the tools to do a key ceremony using smartcards. I will explain soon why we needed to do that.

[Oh, as I said before,] Mac OS X has the native drivers for the smartcard readers. It is just to plug and play.

Well, why Mac? Well, because why not? Well, we have it on hand. We don't need to find some laptop or desktop or Windows or any other hardware. We use the Mac because we have it there. It's so much fun to try to run everything there. We try.

But the name servers are not used to running on Macbooks or Netbooks. They're used to running on BSD, Ubuntu, or CentOS. All this software we talked about are available for those operating systems, too, and the tools we are talking about, especially one tool in particular, we don't have available for Mac. It runs perfectly on any Linux [situation]. [inaudible]

Doing it on a Mac was fun, or really fun, just because the Mac is so simple as a simple laptop or a user-friendly laptop. It's easy to use.

This process we tried. It's based on the key ceremony script written by Richard Lamb and documented by Richard Lamb. We are using the [ISO] image in the key ceremony that he published on his website in order to have the tools. We took those tools and copied them to a USB dongle, a USB key, and ran it from Mac OS X, and almost everything runs perfectly.

Then we started digging a little bit in to the scripts and we figured out everything very, very fast. We'll show you some.

As you know maybe, but I will explain it a bit, when you generate the keys using the smartcards, the generation of the key by itself, the use of random number generator, everything of course is inside the smartcard in a cryptographic device – that's one way to say it. It's not a cryptographic accelerator. It's a cryptographic device. None of the

cryptographic information will be copied or reside on the computers that are running this.

It's the same concept or the same base as the HSM. You have the important, critical data. The key you will keep in a device that is very secure. These smartcards have some very good level of security, anti-tamper, and it's a very good improvement compared with the keys on this operating system or a computer.

If you have the key in the smartcard and you want to use that key to sign your zone, you need to move that key physically to the server on which you're running this. If you server is stored in a secure data center, then all the processes start to be very secure, even [auditable].

What we did was check all of Richard Lamb's scripts into single script, just to demonstrate how a key signing ceremony could be transformed in a 2-minute script. Well, obviously, it's not for auditing, but just to show the simplest technology behind this – not simple, easy-to-use, really. What we have now is easy to use.

It's about 50 lines of code, the Bash code. We used dialog to display/modify environment variables that I think would be a good improvement for the key ceremony because right now there are always other things are running, typing commands from the [inaudible]. Using dialog, you can just ask for the values of the variables.

These are the variables we need to set in order to run appropriately the scripts from the key ceremony [DVD]. The date is constructed in that format. It's not necessary to be constructed in that format. You can build it any format you want.

The domain you use for the label of the key, and it is good to be the name of the domain you will sign.

The password: we set Richard Lamb as a password, just to keep him in part of his work.

The path of the binaries: we had download dccom. Dccom is the [inaudible] where all the scripts are stored.

This pin, PIN1, is the six-character pin of the smartcard. This pin and the SOPIN, two-variables below, belongs to the smartcard. In the scripts, those things are set fixed in the script, but it is important, in order to have these, to know that you can modify these pins. It's better to modify and save it because as soon as you lose the SOPIN, it's like you lost the [PUK] of your cell phone SIM or something like that. It's good to have it.

Any time you can clean the smartcard, remove it, scratch it, and create the keys again, the strip and you slide the smartcard through the keys again. So you slide the smartcard, create the structure, and create the keys.

When you create the keys again, you need to set to SOPIN and the PIN. The PIN will be used every time you use the smartcard or the key to sign. Every time you use the sign, you will need to input the PIN, so this is important to keep someplace safe.

Here we did implement this, but it could be possible. If you run this on a Mac, maybe you could save the SOPIN and PIN, not in environment variables in the operating system or in the script, but maybe in the keychain of the Mac. That way it will be in a very safe storage.

The CKALABEL is made with the KSK. It's the type of the key, the domain environment variable, and the date. In this way, it's constructed the label of the key.

EBERHARD LISSE: As far as SOPIN is concerned, for ease of use, we only used one. But if you make an auditable thing, then you would have two or three, and then give copies of each SOPIN to two individuals, so you need to have three different people available, but not necessarily the same. So if two people have the same pin and one of them is not available, the whole system doesn't fail you; you can ask.

That's the way the root thing is implemented. Basically it's a part or share of a key that is distributed partially to individuals, and several individuals have the same. But for the proof of concept, we didn't set that up because we didn't want to complicate the matter.

LUIS ESPINOZA: Now we start to view each of the commands necessary to create the keys inside the smartcard to use them. These commands, sc-hsm-tool, comes from the OpenSC package, the binaries in that version of OpenSC. We initialize. This would tell to the smartcard that that would be initialized. SOPIN and the [--PIN] and the PIN will clear the smartcard alone. If you have something there, you will lose it. That will be initialized the smartcard and [inaudible] but with SOPIN.

The second command is running again, sc-hsm-tool. It has, in addition to SOPIN and PIN, a requirement to write again on the smartcard. It mentions this [flag] DKEK shares and the number. These DKEK shares is

an implementation of OpenSC in order to allow a backup of the key. Without this implementation, when you create the key inside the smartcard, it is impossible to take out the key from there, only the public key. You cannot extract from there the private key. This is part of the security.

These people of OpenSC implement a way to create copies of the key in different smartcards. The DKEK shares and the number is the number of parts that will be split to recreate the key in order to [park] it. This way is a safe way to make a backup of the smartcard.

The applications of this is great. Why? How many smartcard readers can you put in the server? I don't know how many USB devices you put in it. 10? 12? 20? I don't know.

If you have a lot of copies of the key you are using, you can split the sing-in process in different smartcards at the same time. Then you will have a cluster of smartcards. Then if you need to sign a lot of signs, for example, then you can create at a very low cost a cluster of HSM-based smartcards. You can build them onto there. You can build something. That for me was very interesting.

The other application this copy of smartcard has is, especially for DNSSEC, you cannot rely on a key that is unique and attach it to a device. If you lose the device, you will lose the key. That's not the point.

The idea with this is that you can have four – of for example, .na has six copies – and two of them were going to the data center where the server is stored, one for in-line signing, and the other one stored as a

backup but within the safe facilities. The other two go to some safe at the bank in Namibia.

I can't remember where all the smartcards are going, but the idea behind this is you can create many copies of the smartcard. Handle with care – handle very, very safely – and in that way, there's a suggestion to use this box as tamper-evidence box. Then you can open it without the evidence that somebody opened the box, and keep the smartcard in the – you say no?

Okay. It is possible to open then. Yeah, contactless. Okay, that will be fine. Okay. Then the idea is that this concept of sc-hsm-tool moves to another level the use of smartcards for these applications.

I can tell you for example in .cr there is small ccTLD with 15,000 domains, more or less. In that moment, there were something like 250 signs because not all your domains will be signed, obviously. You will need only 250 signs. If the smartcard takes, I don't know, maybe two signs per second, that would take something like 75 seconds. It's a little bit more than one minute. It's not an issue right now.

But if at some point you need to sign 2,000 records or 5,000 records, then that could start to be an issue for the smartcard. But using this model, you can create a cluster of smartcards and split the sum in pieces and sign each piece of zones with a different smartcard that has the same key.

Or even better, you can have a [inaudible] in one smartcard and a [inaudible] key in another, and then you will have both keys in the

hardware, not in the same smartcards, but different smartcards, or have to have a backup of them, for example.

This was very good news for me, at least, because it enabled this technology for very good applications. Device key encryption key: a very nice thing from OpenSC.

With this command, the dkek-share-2, I will tell them that we will split the key in two parts. This only creates a structure inside the smartcard. Here is where it is really regenerating the parts of the key. Then when you run into it the first time, you will indicate with the [flag] create-dkek-share and will use the DKEK share that is made of one, .pbe. This is the part of one of the generations of the key. Then we provide the SOPIN, the PIN, and a password for that part.

Then as you say, it is two parts in the past command. Then you need to run again with the second part of the key. The idea behind this is because you can split that part to a different person, and you can ask for each person to provide its own password to that part of the key. In that way, the idea with this is to be auditable.

You need those persons to recreate the key and make a copy of the key. If they're not there, it's not possible. I think it would be following some of the real key ceremony, splitting the key parts, and have different persons having a password for each part.

After [2 DKEK] command, the key is created based on these two shares in the smartcard. Now the key is created and then you run, again, the tool with import-dkek-share, and you import the two parts of the share,

providing the password for each part, and that copies the key inside the smartcard. Once there, now you can start using it.

LUIS ESPINOZA: Until here, we can run some Mac perfectly. These are the zone signing keys. In this example, those zone signing keys are created, and so forth, assuming that the KSK will be in hardware.

This is a sample DNSSEC key generation using dev/random. In the example of Rick Lamb's, he used the [inaudible] inside the smartcard as a dev/random for generating these software keys using an [inaudible] in hardware, which is good.

In the example of Richard, he generates two keys. I think it's to have a backup and rollovers of the zone sign keys. I'm not sure why, but it's there.

Once we have the key inside the smartcard, the software that signed the zone uses the public key in the software and the private key from the key. Then you don't have the file .pub for that key. The script will give you some error.

With this command from pkcs11-tool, this is not part of the Rick Lamb ceremony script. It's more like one of our research. We can extract easily the public key from the key and generate it.

What we demonstrate in this case is we can create the keys or copies of keys in a Mac, for example, and then take the keys to the server, and inside the server, generating private key, and then start using the key

for signing the zone. Then it is not necessary to transfer any file to the server. Just move the key physically.

With these other two, as part of the OpenSC, too, the pksc-15-tool -D, we show all the keys inside the smartcard. We'll show you live soon.

In order to make copies of the key inside the smartcard, this is the command you need to use with the [flag] wrap-key. The label of the key, .wrap, creates the wrap. The key reference is the index of the key inside the smartcard and the PIN. This creates a wrap ready for creating copies of the keys in other smartcards, in [inaudible] smartcards.

After doing this, then you can run again the command to initialize the smartcard and create copies of the smartcard.

Okay, yeah. One of the things we found is we were almost running everything on Mac, but when you need to use the key to sign the RR set to generate the DNSSEC key RR sets, the Rick Lamb scripts used a command that is the pkcs11-backup. That command is a binary written by Richard Lamb because he doesn't rely too much on OpenSSL libraries to use the pkcs11. The pkcs11 is native supported in BIND software. We think it cannot handle very well the smartcards. It's made mainly for regular HSM, for one brand in particular.

This pkcs11 backup is the software used to generate the RR sets for the [SS] keys. This pkcs11 backup has a configuration to use the smartcard to read the private key to generate the keys. This is because here is the -F is the label inside the smartcard, -S is the slot, and -P is the PIN of the smartcard.

We tried to compile these in Mac. It was not possible. There's a set of libraries we need to set up there. But we think we'll have it real soon now.

Not issue really because the servers are running Linux and the pkcs11 backup will run on Linux for sure because it's running on Linux, and the generation of the keys and the copies of the keys will prove that we can do it on a Mac. The idea is only to portray the functionality of this.

EBERHARD LISSE: You don't have to go [inaudible].

LUIS ESPINOZA: Okay. This is in summary the list of commands to create the backup key from the wrapped key.

EBERHARD LISSE: We'll demonstrate.

LUIS ESPINOZA: We will demonstrate that, in a way. The real reason why I came to Namibia to work with DNSSEC, Eberhard will explain.

EBERHARD LISSE: If you look very carefully, you see a lioness. It was in the so-called Erindi Game Reserve, which we visited for a day and a night. That's about seven-and-a-half meters from our Land Rover, where we were sitting.

If you look carefully over here, here is her husband, who is munching on the warthog that she killed, and she is waiting for him to finish.

In other words, as we said, Namibia is a place to visit. Namibia is a place to have ICANN meetings, we hope. I for one find myself in agreement with some other parties in Namibia that we should apply for it. We will do so jointly, and I'm quite sure that other people will realize how much fun it is to come to Namibia, not only for the ICANN meetings.

What we will do now is switch this, and I will share my desktop and we'll see whether this works. So now I must share this with Adobe. How do I share this with Adobe? Just tell me what button I must push.

As we said, this is now my desktop. I've used the terminal screen and enhanced the fonts. I hope this is easy to read. I don't want to run too many scripts and explain each individually because it will take too long. The script will not really explain much. We will just see the output.

Okay. You see that this card is being recognized. This is Identive Cloud 2700 and I must say something about that. This is a SC331, which Richard Lamb donated to us. I think he has about 1,000 of those from a company called Identive. When we tried to get those, they're not manufactured anymore. Then we tried to get the next one, which is a 331. We contacted Singapore and they said, "No, this is also not manufactured anymore."

But we managed to get them to deliver to the hotel ten of these at the cost of 30 Singaporean dollars each, which is not that expensive. It's actually fairly cheap. It's about 20 Euros. It's a big cheaper than in Germany in well. So I must say – and I'm going to write to management

– that I was very much impressed how, when we explained to them why we were using this, how quickly they came to our assistance and delivered it against cash. They didn't even bring a credit card machine. They delivered it to the hotel. Nigel Roberts was present. He handled this for us.

I will end this command a little bit later from the virtual machine on the Athena card, which is – I can do this now, actually – on the same card on this.

What we have done is we have taken these scripts that were provided with an [ISO] image from Richard Lamb, which is from a CentOS version 6 LiveCD. We then connected the USB stick to the machine and sort of figured out where the scripts were copied to the USB stick and then worked on it on the Mac.

That allowed us [rather than] use Virtual Box to download a proper version, 6.6 of CentOS, install it, and get the programs in there. Virtual Box has a facility where you can sort of capture USB ports, which are connected to the Macintosh, but you can basically grab them like in this one, where you see on the top left that the Athena device has a little tick mark in front of it. That means it's not visible to the Macintosh. It's only visible to the CentOS.

As we are going to cheat a little bit, we wanted to have two different card readers that we don't get confused which one is which. Also we also, which I found very important, wanted to try two or three different card systems. We are going to put two different cards in our servers, one in the server room, and one in the safe at the data center so that if we develop an issue we have a different hardware to try.

Now, if you put this card in and run this card again, it's easy. If I run now – what is it? Pkcs – 15 minutes debug. Okay.

This is the debug tool, the dump tool, that we said you must dump the card in and you can see that there is a key in here. It doesn't matter. We are going to run the whole stick, which will initialize the card and that means erase the old keys and whatnot.

So now I've written a little share script, obviously appropriately named doitall.sh. It uses dialog. We mentioned that's a tool that's available on Linux, and hence also on the Macintosh, I used Homebrew. I like Homebrew as a distribution of unique tools that I want to install. You can us MacPorts. You can use Fink. It doesn't matter. It's a matter of flavor of the day, and this month it's Homebrew for me.

Then you basically can change everything that is written here, but in the share scripts, the environment variables have a value, and if the value is present before the dialog runs, it does displace the content. When it's done, it saves the changed content in these environment variables and then continues with the share scripts.

I find this very convenient because if Rick Lamb has got ten share scripts, and for every share script, you have to type in the PIN code and the passcode, that makes it probably more safe. For example, if you use two different [S Opens], two different shares, then you must have it so that you have two different people entering their passwords separately from each other and independently from each other so they cannot mix this up.

If we run this, it sees the reader. It's already good. And it now runs through all the commands that we have explained in detail, one after the other. You see it's safe to share -1[pb], and now it does this for the second one.

Then it imports it, and it says there is two and one is still missing, which is also cool. In the second command that we started we said we would have two [DK] shares. Only two ZSKs have been generated, and now it does the KSK, which takes a little bit longer.

These are the commands strictly taken out of the little manual that Rick Lamb has developed. The website is listed. You can go there. If you click on it, it even has a [tag]. It goes straight to the cookbook that he has. We have taken the programs out of the share script and one-by-one and put it in one thing to just make it run.

As you see here now, again, I can convince you if you look carefully: it's a different KSK. It has also in this directory created this here, this wrap file. We need this when we want to later create a backup of that key of that card so that we have different cards with identical keys in it.

Now comes cheating time. I must put it on this one and use the CentOS. CentOS you can run in Virtual Box seamlessly. That means the desktop of CentOS is not there, so it appears almost like a window under the Mac. If I was demonstrating this to somebody less knowledgeable, I would just do this without explaining everything and nobody would be the wiser.

If I now go to this and we dump this, it dumps this with the same key. I'm going to copy this again because I need it, and now when we use

this one share script that depends on pkcs11-backup, which we haven't managed to compile but we have got two offers to look into this, one from Rick Lamb and one alternative solution that we have found somebody who seems to have student who has some time on his hands and is going to have a look at this.

Now we give a passphrase. What we're doing now is assigning bundles that can be run on a name server without a physical card present. That's the way Rick Lamb set it up. We only showed us that we achieved an identical design that we achieved with his key signing ceremony.

I would envisage that if I used this in production, I would not do it like this. I would keep the key on the machine and every hour run this thing with DNSSEC zone signer, taking the key live off the card. That's a matter of debate. We haven't really discussed it with very experienced people who do that, but at the moment, I think we would use it live from the card.

Let's take a password. Okay. Seems I'm going to the right directory. Then I must go and control + paste this. The PIN as we said is 123456. And it successfully on CentOS with this card, which is blinking, generates these key bundles with the pre-sign key. It's basically a [tape archive] that is encrypted with OpenSSL or something there or signed with OpenSSL. We can take the keys out. We experimented with that. It works.

Since this would only run on an actual Linux name server, we didn't really spend too much time on trying to get this to work on the Mac. That's a lower priority. We'll get this later.

Now, we take this card out, keep it here, go back to this and take a fresh card, put it in this machine. Which one must I use again? Okay. Card, yes, and then pkcs15-tool -D. Nope. That is one that I've played with already. I want to take one that is empty. It has got PIN and SOPIN. I don't know how I did that.

LUIS ESPINOZA: It's initialized.

EBERHARD LISSE: Hmm?

LUIS ESPINOZA: It is initialized.

EBERHARD LISE: It is initialized. I was hoping that Rick was sending us some cards that had not been initialized. It doesn't matter. There is no KSK in there. There is only the smartcard HSM definition, the serial number, the PIN and the SOPIN.

Since we took Rick's script, we assume it's the same as [open]. Let's see what happens.

So far so good. It seems to work. I just wanted to remove this card from the system that we see is the only card is connected. This is the active reader. We see now here KSK, today's date, and then 082420.

Let's remove this and put this card in and again [dump] it, and we see this is the same KSK.

Now what we have shown is we have got one original card and the backup that can be used identically for the same process. So if I was now getting ready to get this into production, I would take the other four cards we have here and generate enough cards that we have.

So now that wasn't too difficult now, was it? We haven't really gotten this to work on Ubuntu yet, but that's not a problem. It works on CentOS. It works in production on CentOS because Rick Lamb has got it in production on some systems.

So it's just a matter of how we apply ourselves, and the difference between Ubuntu and CentOS is very, very, very small as far as the actual programs on it are concerned. If we find that the BIND version is not advanced enough, we can talk to Vicky Risk to make sure that they make us a package that we can install and then keep current, but I'm quite sure we'll find a version that is current and that has got the libraries that we need installed so that this works.

Then it basically remains. Since my provision [script] also use DNSSEC zone signing, we just need to adapt the script very, very slightly to instead of using the software key, pushing this into the card and the card does the signing.

In our top level zone we have got 500 names, so three minutes. We pushed the zone and sign it every hour, no problem. At some stage we will think we have done it on every two hours, but it really doesn't make a difference, so we put it every hour. If it was an issue, we could do it

every two hours. We could put two card readers in and sort of do it so that we extend the time.

Diego is wrong in saying that you can use ten cards. If you're starting to have the need to use more than one card, you will get enough clients who have to pay for the service that you can afford to have the hardware. It's not one HSM. You need three. You need two on the site and one in a safe somewhere in case there is a [plane] accident right there and then and you need to get your system within a day or two. So it's not just 20,000 for the cheapest one. It's at least 60,000.

We intend to put two cards in the on-site, and another card to the management of data center, but they must keep it off-site. Then we will keep two cards for ourselves and one we will hand to the Device Chair of the GAC – let's put it like this – who happens to be at the Ministry of ICT of Namibia so that we get some support for DNSSEC signing from them already. They are very keen on doing this, and we also want them to be involved in the system.

When we put this in further production, then we will make it so that they get one of the key shares so that they're one of the key security officers or key officers or security officers – whatever the terminology is – so that we got them not only them involved but we also get a little bit more authority that the [inaudible] ministry and their technical people are satisfied that this is what we're doing and the auditors are satisfied that's what we're doing, and they are participating in the ceremonies to make sure that all is as we say it is.

Our government domain is not signed yet, and the Ministry of ICT is not dealing with the government, .gov, .na . That's the prime minister's office. The prime minister's office is responsible.

But we hope when we get this involvement together – and he's sitting there, so I'm not only saying it for his benefit – the idea is that we get more involvement and that then the acceptance of other departments in the ministry will grow.

Good. Any questions?

Warren earlier said you could in theory read cards when you have them in a tamper-proof bag. That may be so, but it's not an issue for us because we don't really have an audited procedure with different. If you have the card, it doesn't matter whether you take it out or not. The card sits physically on the machine.

Actually I'm not even giving it for transport. I put it for transport in this one. I put it in a transparent one, which is less secure, because I don't want Customs to start seeing what's going on. If they see what it is, they're less likely to as it to be opened. If they open it, then they open it. We write a letter that it's opened, and that's fine. But the point is, generally speaking, these things are branded because we couldn't get them from any other manufacturer. They cost something like ten cents U.S.

WARREN KUMARI: If you put the card in one of the little plastic things that you put PCMCIA cards and then put a security sticker over that as well, then it's almost impossible to get through that.

EBERHARD LISSE: I'm not even bothering. We bought a box of these things, transparent, 1,000, for about how much? 30 U.S. dollars. We bought 1,000 of those. I only wanted 50 but they were not selling for less than 1,000.

This is just to show this is a proof of concept. It can be done by a small ccTLD. It can be done by a small gTLD. It's not rocket science. Of course, you need an expert, but I'm quite sure had you not come and I had really been bored and applied myself, I would with some prompting over the Internet and figured it out, and that's the very point.

Anybody in Angola, in Burundi, or West Sahara can do this. There are enough people on the network. I hope you are from West Sahara. The point is, anybody who is available on the Internet can contact any of the ones that are sitting here, the experts like Warren Kumari, the people who write RFCs, like Geoff Huston and others. We're all here and we're all willing to help. I'm not so much an expert. I'm more like the focus to get experts together and to try to organize these things. I'm more like an amateur sitting on the sidelines. But there are enough people around who are willing to help, so the [lethargy] that we see in Africa is not really absolutely required.

There's a question from the floor.

[GIHAN DIAS]: [inaudible] I'm [Gihan] from .lk.

EBERHARD LISSE: From?

[GIHAN DIAS]: .lk, Sri Lanka.

EBERHARD LISSE: Okay.

[GIHAN DIAS]: Yes, I believe BIND is $25,000. It just seems it's a bit beyond many people's budget. But at the same time, the problem with signing a zone is not the 999 or 9,999 small guys. It's the one bank or whoever it is who might get hacked. If he shows that it is due to your negligence, then you would probably be liable in Namibia or other areas.

Let me just finish. I think it's not feasible to have this really expensive, it just seems. At the same time, we might want to be a bit more, I would say, not so much the card itself. The card itself I think if fine, but the procedures around the card we need to be I think very careful that we can show that we have done everything properly and that the private keys are really secure because the whole point of signing the zone is to prove that if somebody goes to something bad.na… No, it isn't? Okay, what is the point of signing the zone?

EBERHARD LISSE: I agree and I disagree. We are not doing this to be able to go to a bank and say, "Don't worry. We're on the job." We're saying, "We have the technical know-how to do in on a level that we can document where our good points and weaknesses are." It's not that we can do it the bank

wants it, but we can document the way that we have done it, and the banks can assess it.

It would not be acceptable to me if my bank accepted this level, but this is the next step. If the banks wants a higher level of security, they must pay the funds it takes to do that. I don't have the money.

[GIHAN DIAS]: I agree, yes. So people should not expect this for free. I definitely agree. But at the same time, if we are just saying, "This is just a trial and we are signing test domain .na," that would be fine. But we are signing .na. That means you are providing a certain level of service to all your customers.

EBERHARD LISSE: We haven't got many customers. Every single one works for my company. The point is not that the three domains we have and my three partners and my own domains are all signed. There's no commercial demand. That's not the point. The point is, how do you create commercial demand? You must lower the barrier of entry.

[GIHAN DIAS]: Oh yeah, I agree with that.

EBERHARD LISSE: I will tell the bank, "That's not safe enough for my money. We can do it for you money, but my money's not safe enough."

[GIHAN DIAS]: Right. So what's a solution we give the bank?

EBERHARD LISSE: But it shows we have the technical know-how. We just need to replace this by a hardware HSM, or not even that. You can use this and use the scripts that Richard Lamb for the key signing ceremony has developed. But it's a step-by-painstaking-step from entering the facility, turning on the air-con, and explaining the egress in case of fire. Every single step is there.

If you then also make sure that the banks are acceptable about the way the keys are handled, then you can use this. I personally would prefer hardware, proper hardware.

[GIHAN DIAS]: Right. But I don't think the problem is with that card. I think that card is pretty secure. It's how you actually manage and [inaudible] that card and how do you do it. It's the procedure which is important, not really the card.

EBERHARD LISSE: Yes, but we specifically wanted to not do this procedure now. We wanted to first establish: can we do this? It's probably not even safe to do it on a laptop because it has got a virus. We need to sort of get this off and under core conditions. Some people say it's not even safe to do it with the battery inside. All these things need to be considered.

But that's not the point we wanted to make here. We want to make the point here that a small TLD in a resource-restricted continent can do this

fairly simply. If eventually the demand will be there – now you got me; I was actually pushing where my director is sitting there that if the government wants it – he mentioned finances – these things eventually need to go together. I am not going to be able to purchase three HSMs unless I have got enough clients who warrant the investment. It's not really to make the profit. We just can't afford it.

[GIHAN DIAS]: No, and buying HSM will not really help unless you have all the procedures behind it. It's just the HSM will be even maybe just as secure as the card, or the card will be as secure as the HSM.

EBERHARD LISSE: As you know, I'm a gynecologist, and I usually say, "One little baby step at a time."

UNIDENTIFIED MALE: Okay.

EBERHARD LISSE: All right. Good. Any other questions? Good. I hope this was at least a little bit interesting.

The next would be Paul Wouters from CentOS, who we have not maligned too much, I hope.

PAUL WOUTERS: Okay. I'm Paul Wouters. I'll try to make this fast. Catch me in the next couple of days if you have questions about any of these specifics after.

Okay. I am the sort of DNS guy/the DNSSEC guy at the Fedora project. Everything we do in Fedora slowly moves if we like it into the [inaudible] rather than [inaudible] Linux.

Some of the things we went and talked to [inaudible] people and we said, "Well, we should really put all our PGP keys of all our developers in the DNS." We beat the [inaudible] people to it, so all of our users who did upload the PGP keys in the system now have their PGP keys in DNSSEC. So you can easily get them using a simple command.

For testing, I tried to put the [d]-command within 140 characters so I could Tweet, so I had to tweak it a little bit. You can see I couldn't use my e-mail address, pwouters@fedoraproject.org, because it was too long, so I used paul@nohats.ca, and I managed to squeeze it in. So clearly this is not what users are going to do to get a PGP key. This is even worse than GPG itself.

I'm not sure why this is fading. I can paste it back.

EBERHARD LISSE: It does that. [inaudible] Just abort.

PAUL WOUTERS: Okay. I'll try not to paste. I maintain a package called hash-slinger. Hash-slinger contains various tools that create hashes of key material for DNS. Openpgpkey is the command that actually that operates on your local

ring and can take a key from the key ring and put it in DNS or take an entry from DNS and put it into your key ring.

There's a few missing features. Punycode I still have to add support for. There was confusion about the root key location that I found out after I released this last week, and I got a request for using different lines instead of very long one-single lines of like thousands of characters.

How does this look like? If you want to get my key after you install the hash-slinger package, the only thing you have to type is openpgpkey - - fetchpwouters@fedoraproject.org, and you get a nice little line that says obtained from DNS, was protected by DNS, and then you get the key. So it's very simple.

Of course you can also do this from a key server, but from a key server, you don't really know who put it there. Someone else could make my key, and if it's an old version 3 key, then anyone can make a key that matches my key ID, and then you never really know if you got the right key.

Now we're using DNSSEC to actually prove that the administrator of the fedoraproject.org zone apparently put in this key, so probably this is his key.

Once you have a key in your key ring, you can use the openpgpkey - - create command. The first version you see is where the output is in RFC form. It's hard to see because of the wrapping, but it's using the openpgpkey word, but not all software supports that yet. So the default mode is to use the generic version. You can see here type 61. This is a way to display the exact same information, but it will work on older

tools, so even tools that are two or three years old will still able to put this very shiny new record in the zone file and sign it.

I believe BIND and [L-DNS] now support openpgpkey, but for instance, OpenDNSSEC does not, so even if you put it in your zone and then want to sign it later on, then it will not work unless you use the type 61 syntax. By the time it's really well-supported, I'll switch the defaults around.

If you're really daring, you can trust the DNS so much that you immediately put it into your key ring. Here's a version: openpgpkey - - fetch pwouters@fedoraproject.org pipe into gpg - - import, and you'll immediately have the key.

You can see openpgpkey - - verify to actually verify that the key in DNS is the same as the key in your key ring. It's very useful. So people who want to do key signings tomorrow at noon, it would be really useful if you can just put it in DNS and then we can very easily obtain our keys.

The next thing we wanted to do is because someone came to me and said, "Well, why aren't we using this for the distribution keys that sign all the packages as PGP?" I was like, "That's a great idea," so then I looked it and for each Fedora version, we used the same e-mail address: fedora@fedoraproject.org. So unfortunately I can't really make a distinction between the Fedora 20 key or the Fedora 21 key. So I have to think about if we can somehow do this for future versions by at least versioning the e-mail address for it so that we now have fedora21@fedoraproject.org or fedora22@fedoraproject.org.

But it's a good way. You at least have another repository that is a root of the DNSSEC root key where you can actually sort of still have a verification of PGP.

The hash-slinger package that contains this command also contains some other commands. SSHFP is to create and verify a host key. It uses basically the same syntax, -- create, and -- verify

TLSA is not a command that creates TLSA records. That's the record that verified web certificates or SSL certificates. I haven't implemented the STARTTLS support yet. If anyone wants to write some Python, feel free to contact me.

Last one is the IPSECKEY command to create the IPSECKEY records in DNSSEC. This will allow you to run a VPN between two machines based on keys that you publish in DNS.

I'll give another reference implementation, another example. I wrote a small Python package called openpgpkey-milter. This is a plug-in that you can install on your Postfix or Sun Mail server and whenever it receives e-mail either locally or from the network, it will go out and look for these open PGP keys in DNS. If it finds one, it will automatically encrypt your mail before it sends it on. It's very easy to use. There are four steps on the slide.

My biggest problem was that it actually really worked. My e-mail goes to a server in Amsterdam that is the public MX server, and then that machine secretly forwards it to my secret DSL server. As soon as I installed this, all my e-mail got encrypted because the forward actually encrypts everything. I read my e-mail with Pine, and the PGP integration

is not so ideal yet. So I made a feature to exclude certain IP addresses and certain names for this.

But don't use it on a production server. It's really a reference implementation. It's more of an example use. What I really want is for clients like Thunderbird to really put this in so that we can have this automatically fetched in our e-mail clients.

Some other DNSSEC experience we had in the last two or three years in Fedora. We brought in the DNSSEC-trigger + unbound combination to get DNSSEC per default. People didn't use it too much, or some people used it and they used to be knowledgeable enough to just deal with the problems when they would show up. They would be at some café and it wouldn't really work, and they know to either restart DNSSEC-trigger or restart Unbound. It wasn't really fully ready for the end user yet. We still want to put this into network manager more natively, but we lack the resources to do so, so that hasn't happened yet.

When we announced to make this the default for the next version of Fedora, we gained a lot of users who said, "If this is going to be the default, I better try it out." Then we got a lot more reports coming in from people. A lot of them were actually from people who run split-DNS, so you have a different internal and external view. There were not so much DNSSEC issues but more issues with running a validating re-caching resolver on your end node. So you would have your laptop and you would be home and you would be able to read certain things, and then you would go to your campus and you would be able to reach the campus-only names, and then when you go back home, it didn't even work anymore or they were still there.

So we got some people who said, “We want the cache to be flushed on every network change that my laptop does so that I don’t have this problem with split-DNS.” Now we had other people who said, “Whenever I switch my networks, it flushes the cache and it does 1,000 queries, and I really don’t want to do that.” Especially when you’re on flaky Wi-Fi at a coffee shop, you really don’t want to start from an empty cache, like Unbound, which is not very aggressively retrying all the negative caches. So once you start losing packets on your Starbuck’s Wi-Fi, you’re basically dead in the water for DNS if you start with an empty cache. So I really want to try to maintain that cache.

Another reason for me is that I have this idea that if I open my laptop with my combination of programs, if you start all the time from an empty cache, that will really give you a user fingerprint of “Oh, he’s connecting to this and this [IM] servers, and he’s connecting to those and those mail servers.” You really can get tracking me based on my DNS queries. So I really do not want to flush the DNS cache if I can.

But other people, if you’re on a campus and you run 100 internal domains that are not available elsewhere, then those people really want their caches flushed.

So we’re not sure how to fix this yet, but it seems inevitable that by putting DNSSEC on the end node we are going to break some people’s network. There’s definitely still some fighting about what can we break and tell them to fix design-wise or what will we have to support. It’s really hard.

If you're running anything split-DNS, really start thinking about if you could maybe phase this out, because it's going to be harder and harder to deploy that.

We did manage to make this split-DNS work for the VPN scenario. For instance for me, for the VPN, if I connect to the Redhat VPN, there are some internal-only DNS entries. Since the VPN transfers both the domain name and the name servers used for that, as soon as you run your local resolver, I bring my VPN up and it installs forwards for the Redhat.com domain. So everything works transparently for me, and it actually specifically flushes the Redhat.com zone when I disconnect my VPN so that I can still reach Bugzilla on the external IP address.

So it works really well for one domain, but neither DHCP nor the VPN protocols really allow for transferring of 100 internal-only domains. So split-DNS is really a problem that no one has really tackled yet.

I'll just go to the next slide. The other thing we've been working and looking at is how to deal with virtual machines and lots of servers and containers. Virtual machines tend to have their own infrastructure, so you boot a whole full machine and it can start a resolver and it can start lots of other things. It's a lot of overhead.

What you see now more and more with things like OpenShift and OpenStack and SELinux name spaces and containers is that all these virtual machines are becoming smaller and smaller and they're turning more into an isolated process. You don't really want to run a caching name server to do DNSSEC validation for all of that for each and every one of them. So if you run 1,000 containers, you don't want 1,000 Unbounds running or 1,000 BINDs running.

How do you deal with that? Ideally you would have something like running one of them for a cluster of virtual machines or one Unbound per [inaudible] that you run, but then you run into the risk of: are you going to trust the AD that it technically comes from the network, even if the network is just another VM on the same hardware?

People have different opinions about whether or not you should trust that AD bit or not or whether it should be configured to be trustworthy or whether you really don't want to trust the [external] network and it must always be validated on the host, or maybe even in the application itself.

For instance, the Libreswan IPSec software right now has the root key embedded in its source code and will do validation and using its own cache and just uses the system DNS servers as a forwarder. So it will get everything from cache, but it will still do all the validation itself. That's a security decision made because this is a VPN; it's a security software.

But people don't like that if every application will start doing that because then you're building up all these DNS caches and again you got lots of copies and data and you can run 1,000 servers on one hardware node.

Another thing that we need to start looking at is the planned root rollover for the KSK. If you have software or distributions that have this key hard-coded anywhere and are starting to roll this key, then we better update the software. And we have to make sure that if someone pulls out a Redhat Enterprise server off the shelf in ten years and turns it on that it's not dead in the water until they manually fix something.

So we need to figure out, when the root key actually rolls, how to make sure that our servers won't break. That's also not a real solved problem yet.

What I'm mostly working on these days is to actually do a better integration of DNSSEC with IPsec. As we've heard through all the Snowden revelations, there's a lot more pervasive monitoring happening, where massive eavesdropping on a massive scale is happening, and we really want to try to deploy VPNs everywhere so that everything becomes encrypted, whether it's within the Cloud, whether it's just two machines on the Internet.

One way of doing this is something that I think goes back to 1999 when [John Gilmer started the Free Zone Project], is to basically doing IPsec based on key material that's published in secure DNS. We're actually working on it right now. For the people who are on the IPsec mailing list, they might have seen me a lot working on the Anonymous IPsec part.

When this was attempted in the past, one problem was that both sides would have to authenticate each other with IPsec, and that's really hard because that means that you have to configure your laptop with some key to talk IPsec. If you compare this to TLS, where your laptop really doesn't have any key, it remains anonymous to the server, but it just authenticates the server.

So that's one mode that we're going to implement as well where you can do this with IPsec, so you as a client will be anonymous but you will be building up a VPN to a server, and then you double check with the DNS record to be sure that you're actually talking to the right server.

That's hopefully in the next couple of months and probably at the IETF, so I will give updates on the status of this code.

Those are the things that I've been busy with in the last year or so. Does anyone have any questions on Redhat or CentOS or Fedora or DNSSEC? Now's a good time.

EBERHARD LISSE: Okay, thank you very much. Any questions from the floor? No? Then thank you very much.

Jay had a conflict with the DNSSEC meeting, so I've asked Stephen Deerhake to give us his impressions as the closing remarks.

STEPHEN DEERHAKE: Hi there. As Eberhard mentioned, I got called in here on rather short notice, so I have been scrambling a little bit to put something together for you.

I think you'll agree with me that we had an interesting day with presentations covering a number of distinct areas of interests. Registry operational issues were covered in two presentations. Certainly the work that's been done by .ph providing a simple API interface [that's register-based] while supporting a number of backend registry systems – specifically FRED, CoCCA and their own Legacy systems – is intriguing.

I do have some concerns about the apparent complexity of their solution. It will be interesting to see how they evolve their setup in the coming months and intermediate future.

Nigel's presentation illustrated an elegant, relatively easy-to-implement approach to host level security using IP sets. I myself have been doing some work independently of him on this.

Besides providing a current kernel level security for the registry applications, the approach also generalizes into enforcing application level usage upon policies such as enforcing WHOIS lookup limits in an automated fashion in a real-time basis. I've been doing work on this myself, and we automatically throw people away and generally torture them if they abuse our WHOIS, but then they eventually figure out that they're back in again and there's no work on our part.

Geoff Huston provided us with a very informative presentation on the state of the address space, as well as some interesting information on trends and routing last year. I think he certainly made the case for greater uptake of IPv6.

I also found his information on the rise of the secondary market in IPv4 addresses very interesting. I think it's inevitable that we're going to see more and more secondary market activity. I was really fascinated by his work showing how old IPv4 blocks seem to be coming out of the woodwork now that they have monetary value behind them.

I think, however, as a community we should have some concerns about the accuracy of the numbering organization's WHOIS databases going forward as more and more of these secondary market transactions happen without the knowledge of the regional registries.

Lastly, I think we all have a pretty good take on his view of home routers.

DNSSEC was a topic of three presentations, ranging from the problems of switching KSK, key generation, algorithms, and a production registry, from Simon Balthazar, .tz, to Luis' and Eberhard's presentation on the work they did with key generation and signing with smartcards for .na, to the presentation that we just had from Paul Wouters.

As an operator of a relatively small TLD myself, I greatly appreciate the prototype work that Luis and Eberhard have done to lower the cost and decrease the complexity of trying to sign a zone.

Victoria Risk of ISC also presented on the issue of the retirement of ISC's DLV service, so I guess we really had four rather than three DNSSEC-related presentations.

In the morning, we also had a very entertaining and informative presentation from Cory Schruth of ICANN on what it takes for ICANN's IT department to provide us with the Internet access, the remote participation, audio streams, Adobe Room presentations, scribing, and all the other IT-related things we have come to take for granted at recent ICANN meetings.

To me, it's astounding what they're doing, and over the last three or four years, I think it's day and night compared to what we used to be getting along with on that. Personally, I think he's having way too much fun doing it, but I can imagine it's a rather stressful week for him as well.

We received updates from Cristian Hesselman, Chair of the ccNSO SECIR Working Group on that group's progress in sorting out the issues involved in establishing a secure mailing list for ccTLD operators. To me

personally, it's kind of astounding that nobody seems to have a reliable way of getting ahold of people, but there you go.

We had a presentation on developing Ipv6-based identity registration protocol and its possible application to implementing true End2End point links.

Our local host is .sg. We had Mon Perez of .sg provide an interesting presentation on how they've employed business intelligence techniques to gain a better understating of their registrants. This is something I've done piecemeal, but nothing as systematic as what they're doing, but then they're much bigger than I am on that.

Lastly, I'd be remiss if I failed to mention that Marc Blanchet presented an update on the status of the IETF's work on RDAP as a replacement for the current WHOIS protocol. It should be noted that in the Q&A for that session, some concern was expressed about the impact this protocol when employed might have on the IANA.

I think that's about it. I think overall it was another well-worth program. I hope you guys enjoyed it. Thanks for coming.

EBERHARD LISSE: Can I have the e-mail, please?

[END OF TRANSCRIPTION]