HYDERABAD – DNSSEC Workshop - Part 2
Monday, November 07, 2016 – 11:00 to 12:45 IST
ICANN57 | Hyderabad, India

JULIE HEDLUND: Welcome, everyone. This is Julie Hedlund from ICANN staff. We are starting up Part 2 of the DNSSEC Workshop. Please come in. Take a seat. We've got some seats here at the table and plenty of other seats as well. I would just say that the next panel on the Root Key Rollover Discussion Recursive Resolver Software Readiness is going to be moderated by Jacques Latour. As soon as we get settled here, I will turn things over to Jacques.

JACQUES LATOUR: Hello. My name is Jacques Latour. I'm with .ca. Today, right now, we're having a panel on the Root Key Rollover discussion and the implication of Recursive Resolver Software Readiness.

I'll start with a small intro. Yoshiro talked about JPRS outreach in Japan to ISPs that the issue around the key rollover that it's happening right now. It's not an issue of "if" but it's an issue of "when."

A new key was generated for the root zone in October. February 2017, a new key is going to be published. It's going to be signed

afterward and then it's going to be used. Then January 2018, the old key is going to be removed. What it means is that everybody that's running a recursive validation, recursive right now, they need to do something. All the recursive software need to support this change. It's happening right now.

On the panel, we have Unbound Jaap from NLnet Labs, Ric from ICANN, Rod Rasmussen on the phone, Mukund from ISC – nice to meet you – and Jaromir from CZNIC. We're going to have from each resolver, how they're going to respond to the key rollover and explain a little bit around that. Jaap.

JAAP AKKERHUIS: Hi. The Unbound is not only a fashion statement but because of rollover recursive server. Next slide, please or do I have to click that?

JULIE HEDLUND: No, we don't have a person.

JAAP AKKERHUIS: Okay. In short, it's a validating, recursive, caching and DNS resolver. This validates DNSSEC since its epoch, meaning it was built with DNSSEC entirely in mind. It's not added later on as a vital on the side. But that's for the Unbound things.

Next one, please.

For the KSK, we actually have support for RFC5011 since November 2009, that's actually since it became official IETF standards, we merely implemented it. Also, it will do things automatically do the rollover necessarily and the option there is the auto trust anchor. When it finds a new trust anchor, there's an awful lot of checks. It actually use that. That's for Unbound.

But since the root got signed in 2010, July 15, 2010, we actually put the anchor in the distribution as well. To make sure that we have proper anchor there and distribution, we actually got a paper copy of the ICANN Cert in a tamper free envelope directly from Marina del Rey so we could do out of band verification for the online anchor.

Up to about one and a half year ago, we only did unit testing whether or not this thing work because there was never been a KSK rollover. High technical but everything seems to work.

Next, please.

Here is the tamper free envelope. I think it's signed by [inaudible] of certificates of ICANN. It's signed by, I think it was [Baxter] and it's pictures, there's videos, yes. The pictures made witness by Olaf Kolkman, one of the trusted community representative, me and somebody else which I forgot. Anyway,

this is what we can do to check whether or not anybody makes a typing error in the security things which goes into the Unbound.

Next, please.

What did we do to prepare for the proper KSK roll? What activities did we do? We actually became active in the KSK Design Team so we could keep close tabs on what is going on and also help with designs with Olaf and see whether or not this works.

We tested against the various testbeds which people set up ad hoc enough and which actually showed it worked but was small problem. Since we actually fall into [inaudible] and the testbeds rolling over weekly, then the standard says you can do. We had to add codes to allow actually testing.

So [inaudible], showed of all of this testing is that we didn't found any serious problem. We did find that some small changes we had to make for hardening about corner cases. But again, these corner cases were actually caused by the first rollover, then act the basic problem was.

If people are just doing the 5011 and using default complication and don't do weird stuff, we don't expect any problems with Unbound.

Next, please.

We're still doing more preparations. The moment the new key will actually become available, which I think is after the next signing ceremony, we will want to do the same out of band verification with the new key. We will ask the new key to the distribution on the moment it is useful to do that and to, of course, keep track of all the discussions which is happening worldwide on what other people are seeing going to do similar stuff. We don't make the same mistake again. That's actually our current plans.

Next, please.

Yes. What do we actually do? There's also an Unbound Anchor tool which does sanity checking of the 5011 methods. That's why we have this cert in the end. We can do in band KSK retrieval and verify the ICANN cert whether or not we actually got the proper anchor.

Where you can find all this stuff, it is in the IANA websites where the cert is but we have our own copy so we could check whether or not IANA make the typing error on the website.

Now, that the anchors we distribute are actually used as hints. It will check against the outsides about new anchors and before we start to use them. If the distribution is, for some way, being

broken or otherwise, it will actually be detected by the moment you install or restarted the system.

Next, please.

The other question is what happens with the old key when it gets revoked. We will remove it from the distribution. That, it will be time to remove it from the distribution. So far, while we actually have both keys in the Unbound distribution. That's about it.

JACQUES LATOUR: We'll take questions at the end. Thanks. Next one is Ric Lamb from ICANN talk about KSK Rollover.

RICHARD LAMB: All right. Hey. Everyone here is, of course, the DNSSEC experts. This is the official ICANN presentation that we're doing a roadshow with. You'll see this presentation over and over again. We're going to keep beating the drums until the key roll happens because the main point here is we just want everyone to be aware this is happening and be ready.

Next slide, please.

Okay. There are not. Everyone here knows what DNSSEC is, right?

UNIDENTIFIED MALE: [inaudible]

RICHARD LAMB: Okay. I don't want to go in to waste a lot of people's time here but I was told not everyone's a DNSSEC guru in here. I'll talk a little bit to it. We're about to roll the key as you heard Jaap say. This is what the point of this messaging is. This is important to ISPs, anyone running a resolver.

Next slide, please.

Okay. Do class exercise. Okay.

Next slide.

The current root key was actually generated in 2010 as Jaap pointed out, and we're very proud of how it was done because, of course, some people have difficulty trusting ICANN. We built a system of 21 trusted big community representatives, some of which are in this room, that hold physical keys, smart cards and other pieces to allow us to do what is called the key ceremony

four times a year where we make use of that key. Without them, we're not able to make use of the key.

Next slide, please.

Those were pictures of actually where the key facilities are. Because this is ICANN and because it's a very open system, we publish everything about it. There's live streaming at every one of the key ceremonies. External witnesses if you're ether in the Los Angeles or Washington DC area, please let us know. If you want to be a witness at this key ceremony, you can certainly come. We'd be very happy to have you.

This was a group effort by the community for the community. If you look in that picture, you'll see Vint Cerf off to the right. Of course, one of the things, if Dan's online, he'll notice it. Dan, you were always embracing DNSSEC so we embraced you. He is actually one of the 21 people. You'll see him there. Dan Kaminsky is someone who actually helped DNSSEC take off by making very public some vulnerabilities in the DNSSEC.

Next slide, please.

Why change something if it ain't broke? Its secrets don't remain secrets forever. I don't know how many people in here spend time with Cryptogeeks. But even at these meetings that I occasionally go to where you see people at Whitfield Diffie and

others, there's always a little uncertainty and depending on who you ask, how long is this key good for? That's the question.

If the key is good for six months, you got to change it every six months. But no one has an exact date. It varies a lot. That's one of the reasons why we say we should change this. There's also always new discoveries that happen, vulnerabilities to crypto algorithms. Right now, we have a 2048-bit RSA key for the root. We're going to continue using a 2048-bit RSA key.

The second reason is if we don't ever exercise this, rolling the key when we have to do it, we will not know how to do it. It's very good to exercise this.

The last one, to me. Is the most important was that we did promise the community in the original documentation and setup for this that we would roll the key five years.

Next slide, please.

All right. It's going to impact a lot of people. One of the reasons we're talking about this and it's so important is if we screw this up, there's a potential for affecting 15% of the people worldwide that are using DNSSEC validation. The reason it's about 15% is, thanks very much to Warren's company, Google and 8.8.8.8. Everyone in here knows 8.8.8.8? Okay. A wonderful effort by some gentlemen in the Google offices in Manhattan in New York.

A lot of people use it. It's got validation built in it. If you want to dig deeper into that number, Geoff Houston has come up with some really interesting approaches to getting this information and doing research to come up with these numbers. That's where we got those numbers.

Because this is going to affect such a large set of people, we've taken a very slow approach, very careful approach to coming up with a plan for rolling key. We've even come up with back out plans, fall back plans as well. We're doing this in very small steps.

Next slide, please.

Proof that the pudding is right here, if you go to that website, all the plans that were developed with the help of the community, I think some of the members of the Design Team, the community Design Team are here. I think Yahoo! is one of them, definitely.

This is the result with the staff took from the Design Team recommendations. Please, if you have an opportunity, it's not that heavy reading. But if you see any problems with this, we would much rather see that now than later.

Got it. No, I'm sorry. Just mosquitos. Sorry. Please take a look. Next slide, please.

That's not a good picture. I sent you a fresher, an updated PDF, could you pull that up?

JULIE HEDLUND: No, there's not enough time to do that.

RICHARD LAMB: All right. Thanks a lot.

JULIE HEDLUND: We'd definitely [inaudible] in the room.

RICHARD LAMB: Okay. All right. We just generated, as is pointed out, the new key just a couple weeks ago. You'll see in the right there a hash with the DS record of that new key there. We all signed the sheet of paper. We're all proud of that.

The picture on the left would have had all the people at that key ceremony and you'd find some members that you know there as well. But this is there just proof the process has started. This key is not visible in the DNS yet. But it has been generated. It has to before we can actually say it's good, this key has to show up at the backup site. If you think about it, a key on one site is no use

unless it's on both sites so we know, in fact, there's a backup copy of it.

Next slide, please.

There's some important dates to watch here. The size of the DNSKEY RRset set record is going to increase in September 19th. It'll go up to 1414 bytes. This is one of the things we looked at very carefully and we've done a lot of test but also monitored where we could see from the experience of other TLDs that have gone through this process before and increased the DNSKEY set.

We don't think it's going to be a problem but this is one of the first times. In September, you'll see a change. In October 11th, that's drop dead. That's when we change the key from one to the other. If you're not there then, you're going to have a little bit of an issue at that point. Okay. Almost done, I got one more. Okay. In January 11th, we actually evoked the key.

Next slide, please.

That's the picture. I encourage you to look at it later. It tells you everything.

Next slide, please.

How do you do this? RFC5011, we talked about that already. We're doing a lot of testing with various setups. There are other programs that work with Unbound Microsoft Resolver as well.

Next slide, please.

Here are some sites. Warren Kumari was kind enough to develop something, key roll systems. Yours truly did something there in the [root] servers. These are accelerated testbeds. They will do the whole key roll schedule, the whole schedule as planned within a very short time. We're also going to have real time testbeds available as well very soon. They will be show up on the e-mail list.

Next slide.

That's it. This last slide shows you the mailing list you should subscribe to to hear some of this stuff or keep track of this stuff. But please, if you have any questions, do not hesitate to contact any of us. Actually, many of the people in this room who were closely involved with this process and you will get directed to the right person because we want to hear if there's any problems or concerns people have. Thank you. That's it.

JACQUES LATOUR: Thanks, Ric.

RICHARD LAMB: Yes.

JACQUES LATOUR: Next one is Rod Rasmussen. He's Infoblox. He's on the phone. Hopefully it's going to work.

ROD RASMUSSEN: Yes. Can you hear me?

JACQUES LATOUR: Yes.

ROD RASMUSSEN: Okay, great. Now, you guys are coming through loud and clear. I'm getting an echo, unfortunately. But I just will quickly give you two points here on Infoblox and what we're doing. The Infoblox is a company that provides DNS resolution services in our products which is sold primarily to large enterprises and ISPs, etc.

We are also dependent upon BIND as our underlying technology. The IRC portion of this [Taco Pro] would be more informative on some of the technical details from an actual… That was really is

getting there. Sorry about that. From an applied experience perspective, we have a lot of enterprises. We've been doing KSK rolls on our own zones for quite a while in various forms and functions. We have a lot of experience dealing with all the problems that have cropped up in the past.

We're not anticipating any major issues with the upcoming roll. We have done a fair amount of QA type testing in our own labs on various types of scenarios with different configurations that people might be trying to run. We have a customer-based instead of pretty diverse set of requirements for doing DNS resolution and running our own zones in a kind of split environment as well.

So far so good on what have been coming up for potential issues whether KSK roll for the root. There will be some more intense testing obviously once the new root is published. But our client, at this point, is basically to take that in February and run with it in a more intensive cycle and then, of course, work with IFC as our partner to make sure there's nothing else going on that we need to be worried about.

But our experience has been that it's not too much. Thank you whoever fixed that problem whether [inaudible] actually talk. I was going to say just finish up here by saying that we don't

anticipate any major issues with the roll based on our experience so far. That's all I had to put out at this point. Thanks.

JACQUES LATOUR: Perfect. Thanks, Rod. Next, we have Mukund from ISC and it's your first ICANN, I believe.

MUKUND SIVARAMAN: Yes.

JACQUES LATOUR: Welcome.

MUKUND SIVARAMAN: Hello, everybody. I'm a BIND developer so I'm going to talk about support for the group key rollover and BIND. As you know, BIND has a validating resolver implementation. We have had it for ever since DNSSEC was ran I suppose.

Next slide, please.

Okay. For the validation to happen, you need a starting point. That's where you introduce trust anchors. Validation happens as a chain, basically a chain of validations. The starting keys, initial

trust keys provided by an administrator or as part of the software itself.

In this file called bind.keys provides the initial starting trust anchors. These are static trust anchors. These are unchanging read-only trust anchors. We'll come to that later.

BIND, as a resolver, can turn off DNSSEC validation when you don't need it. You obviously want it on. When it's on, it can either get the administrator to statically manually configure trust anchors or for the root zone, it can use built in trust anchors or it can load trust anchors from the bind.keys file which we provide again as part of the distribution. This is for simple configuration.

Anything that is non-root obviously is possible to have other roots, other security roots, other starting points for trust. For these DNSSEC, we will have to configure trust anchor, set them up manually.

Next slide, please.

There are two ways of configuring trust anchors. One is configure action called trusted keys. These are static. Whenever the trust anchor is changed, they need to be manually updated. The other way is to use configure option called managed-keys and that's basically the RFC5011 implementation.

The contents of bind.keys actually, it's a configuration snippet basically which introduces the initial configuration keys, initial trust anchors. It also sets up managed keys or RFC5011 for the root zone. Again, RFC5011 is about maintaining trust anchors once they are available but RFC5011 doesn't provide the initial trust anchors. That has to be manually provided or provided by the software itself. bind.keys is not.

Next slide, please.

Okay. This RFC5011 feature was introduced in 9.7.0. It was introduced by my colleague Evan Hunt. It's called managed-keys in BIND. We have a bunch of system tests to test this feature. In BIND, they keep running all the time.

Last year, some bugs were reported including a crash bug and these were fixed. We don't know of any other bugs in that area. We expect the key rollover to go proceed smoothly. Not key rollover.

Now, as I said, bind.keys is the initial trust anchor. Basically, when a key rollover happens, BIND has stored a new key somewhere. It creates another zone file called managed-keys.bind or viewname.mkeys which is actually a zone file which utilizes the keys that knows the RFC5011 keys basically. This can

be in different stages depending on when they were introduced, etc.

The main key point here is that bind.keys is a read-only file. It is not updated. Once a managed-key is configured, the keys are maintained and managed-keys.bind or viewname.mkeys is still part of that view.

Bind.keys can become obsolete at some point in time when a rollover happens. For example, if you look ahead two years in time and the root key rollover has happened and an administrator starts using BIND, basically, he starts a copy of BIND that was built today, he's going to start with a stale copy of bind.keys. The resolver that's configured with such a copy of bind.keys is not going to be able to start validation from the root zone.

It's important to keep bind.keys up to date because the RFC5011 doesn't provide initial trust keys. Any resolver that doesn't observe the rollover happening and misses the rollover is going to need an updated bind.keys. In any case, it's good to have an updated bind.keys when starting a new resolver.

Next slide, please.

In implementation, as I said, when the rollover happens, we store managed-keys in a zone file. We basically have a hack. We

store it as a private user RRTYPE 65533 where we store the various DNS key fields and the RFC5011 fields like the hold on, add, remove and refresh timers.

That, again, if we will implement this from scratch, again, we probably wouldn't [realize] it to a zone file. There are no problems because of this but we realize after implementation that this is not the nicest method because you can imagine what happens when you use 65533 in implementation and somebody else wants to use 65533 as well.

Again, we've looked at all those issues and that's not going to affect anybody but just to point an implementation. Again, I want to restate that it is very important to have a current copy of bind.keys. Always have a current copy of the initializing keys when you start a resolver instance for the first time because the resolver may have missed the rollover, a previous rollover.

Next slide, please.

Okay. That is one recommendation. The other one is you can actually test your copy of BIND for basically root key rollover by visiting Warren's website, keyroll.systems. There are also system tests which you can look at and modify as part of the bind treats of. That's it. Thank you.

JACQUES LATOUR: Okay. Thanks, Mukund. That was pretty good. Next one is Jaromir from CZNIC and he's going to talk about the Knot Resolver.

JAROMIR TALIR: Hello. I'm from CZNIC. My name is not Andre. Surprise.

Next slide, please.

I will talk about Knot Resolver. As my first messages that if somebody says open source DNS resolver, it's not about BIND and Unbound. It's also about Knot.

We have released this new software this year so I will quickly summarize some of main features and new features specifically the feature released in August. [inaudible] and I will also mention how Knot Resolver is prepared for a root key rollover which is perfectly the same as Fox from BIND and Unbound teams.

Next slide.

Knot Resolver, it is a new open source DNS resolver. It's based on Knot DNS libraries from alternative Knot DNS server. It was released this year in May. There was one more version released in August. We have a fancy website where you can go and look up for documentations, source codes. We also have binary

packages for Debian based distributions and RedHat based distributions.

We also use this resolver for our open hardware project called Turris Omnia where we build [inaudible] of routers that we are right now distributing. There is more than 4000 of them deployed. We expect a feedback from the users even about the Knot Resolver features from this group of users.

Next slide.

Just some features from Knot Resolver, it has a flexible cache backends for persistence. The cache survives even as resolver reloads. There is a possibility to configure a cache backends either local or remote to memcached or redis. It's possible to start a new instances of Knot Resolver to connect to this backends and immediately start to serve data from those caches.

Regarding the performance, there is support several application. You can launch many, many copies of the instance. There's no necessity for internal threading. It has quite low memory conception, thanks to lmdb library.

We have done some performance testing. There's a link on the slide. You can go and see that we are a little bit better than BIND

but not as good as Unbound. We are working on that. Of course, that we support also Happy Eyeballs for IPv6.

Next slide.

Some of the features, the whole server is written in combination of C and Lua and it's extensible by writing modules in C, Lua and Go as well. Actually, we have been the first resolver that support QNAME minimization by default. This is the DNS privacy feature. Also, we support DNS64 protocol or technology to supplement NAT64 for easy IPv6 Transition. There is a really interesting way how to filter all different traffics based on views and ACL. We have a quite powerful querying or filtering engine.

Next slide.

In the new version that we released during the summer, we implemented also DNS or TLS as another way how to improve the security of DNS and also DNS cookies which is another technology to fight against DoS attacks by authenticating DNS servers.

We have a new HTTP module so you can immediately see or immediately query what's going on inside a resolver. There's even more powerful DNS firewall if you want to know more about specifically these features. My colleague Andre presented

that during the last five meetings so there's a presentation about that. The link is to on the slide.

Next slide.

Coming back to DNSSEC, of course, we support full DNSSEC specifications including negative trust anchors and including new algorithms based on electric cryptography. There is a small issue with implementation of checking disabled flack that's in progress which I guess would be released soon.

Next slide.

For the root key rollover, we do pretty much the same as BIND and Unbound. We implement RFC5011. Running instances will immediately get new key. One thing you should care about is to have the key file permissions set up to be file as writable. We also have those Debian and RPM packages that contains existing key. We, of course, will update these packages when the new key will be published.

Next slide.

If you instill the server from the source code, there is a feature of downloading the actual current valid root key from the IANA sources. For this to be able, you, of course, need some functional DNS resolver because the IANA didn’t want to fix the IP address

of this source so you need to have the DNS resolver to resolve that URL.

How to validate the response, unfortunately, the Luasec module currently doesn't support the PKCS#7 specification so it's impossible to validate immediately the content of the response. We rely on normal SSL validation so we put a CA certificate of IANA's certificate, authority DigiCert into the source code. The much more information about how to setup DNSSEC you can find out in the specific documentations that you can see on the slides.

Next slide.

That's all. Thank you for listening.

JACQUES LATOUR: All right. Thanks, Andre. Sorry. Jaromir. Okay. Questions, of course.

ROBERT MARTIN-LEGENE: Hi. This is Robert Martin-Legene from PCH. It's not so much a question as it's a complaint because I had a colleague that went looking for this software and he couldn't find it because he went to the original Knot authoritative webpage and there wasn't a link. Maybe he didn't read it.

JAROMIR TALIR: There was what?

ROBERT MARTIN-LEGENE: There's not a link from the Knot authoritative webpage to the other one. It's a completely different domain name. I had to use something horrible called [inaudible].

JAROMIR TALIR: We are going to fix it.

UNIDENTIFIED MALE: This question's on everything or just on Jaromir's? Everything? Bind if you have managed keys and rollover key if you have trusted key. It just keeps the original one as configured. This means that if people have been going to things like the DNSSEC workshop or DNSSEC for beginners and just blindly cutting and pasting examples, a fair number of people might have trusted keys instead of managed keys.

Do we have any idea how many people are doing trusted keys instead of managed keys?

JAROMIR TALIR: I don’t know firsthand but most resolvers are supposed to go configure with the DNSSEC validation auto which means that bind.keys which is the contents of the bind.keys file which is built in into BIND is used by default. If you also provide a bind.keys file alongside it, that will take precedence. But again, if people have trusted keys configure the static trusted keys configured manually. I'm not sure how many of them have.

UNIDENTIFIED MALE: I guess just as a very quick follow-up. A lot of the examples and initial training stuff all use trusted keys because that was all that was available was before 5011 occurred. People who came to the initial training or follow the examples of a [posted] fairly much everywhere online might be configured with the old non rollover versions.

JULIE HEDLUND: I'd just remind people to state your name when you're speaking.

JACQUES LATOUR: Any questions?

JAAP AKKERHUIS: Maybe to start with this one, pitfall in this whole idea of doing 5011 rollover which I actually always going for. The difference between learning a new server as it is and the 5011 is that the standard name server software actually is an install and forget thingy. You just configure it once and it will always be the same until you do this manually.

5011 does actually changes your configuration automatically. If you have taken safeguards like storing all your files on read-only memory so nobody can tamper with your system, you are into a surprise on the moment that the 5011 tries to write out trust anchor because it won't work.

That is actually principal difference in how you operate name servers. You should be aware of doing that when working with the names servers in general and enabling 5011.

JAROMIR TALIR: Yes. Just it's interesting that Jaap mentioned that and it's not, for example, not just about the root key rollover. We, for that he said, we are planning for doing next year also to do our KSK rollover and to change the algorithm actually to ECDSA. This means also that some people will need to go back to their name servers and probably upgrade because not all versions of all resolver support this.

We plan actually to do some campaigns to have ISPs and general public. If they are running DNS resolvers, please ensure that you upgrade to most recent versions during the next year because there are some important events going to happen during that year.

JAAP AKKERHUIS: My question is which platform support the key Knot Resolver? Could you please explain about that? Is it for only the open source platforms or do you have a plan for closed source platforms?

JAROMIR TALIR: I know definitely it's for a Linux. I think if it's the same as a Knot Resolver, if it's for Knot, it's for previously in Mac. I will have to check if we support Windows. But I would say that we do. I think it's for all platforms.

JAAP AKKERHUIS: The same with Unbound, it runs on the standard open source platforms and [inaudible] distributions there. Recent windows binary version so you can install and there's a Mac version and fairly start Mac versions even. The platform is not problem for using these things.

There are other things that you will notice and Rod actually alluded to it a little bit in his introduction. When he said, we actually use based on BIND. You actually find Unbound and BIND and I don't know about Knot, in various proprietary products which mean under the hood uses a version of open source software but with add-on things to further as an add-on service which Infoblox and more of those which Secure64 is one of them as well which actually use combination of things. Under the hood, you probably might find all this stuff as well if you look closely.

There are also some other proprietary only name servers and I don't know what – that's difficult to see what they're doing but they all should be ready for the KSK rollover. Ask your vendor in that case. One of the things in the KSK rollover plan and project is to have trying to contact every vendor on the planet that they actually know about this so they can be prepared if they pay attention.

JACQUES LATOUR: Our last question, Ric.

RICHARD LAMB: I just wanted to add to that. The Microsoft DNS resolver does work with 5011 and it actually works pretty well. I'm one of the

guys testing it inside. I've spoken with them. I know this is an open source crowd so go ahead, throw your tomatoes at me. But just to be fair, they've actually have some really sharp people there that have stepped up to the plate and improved that platform.

JACQUES LATOUR: This concludes the panel. Thank you.

JULIE HEDLUND: Thanks, everyone. Now, I will let Jacques stay here because you're next.

JACQUES LATOUR: Okay. For this session, I'll be talking about some DNSSEC automation especially around DS automotive provisioning. Here we go.

Next.

One thing, I did a couple of presentation in the past around this topic. But essentially, the way to enable DNSSEC or to do maintenance today doesn't work very well with registrant, meaning when you sign your domain, you have a key, you have a cryptographic material that you need to take and bring to your

parent for them to sign. You need to copy a DS record from your zone to your parent. The normal way to do this is cumbersome today. It doesn't work very well.

By far, the preferred method to do this today to sign and to do DNSSEC maintenance is through the standard protocol that we have today. That means you're on your zone, you sign it, you generate the DS record, you give it to the registrar and they submit it to the registry via EPP. That DS record makes it in the zone file.

The challenge with this model is that not all registrar accept DS records. That prevents DNS operator to sign those zone. In the instance that this model doesn't work or is not supported for registries, then we need an alternate way to get the DS record inside the zone so that their DNS operator can sign their zone.

Next.

There's all of the stuff that happened around that. CDS is a new record type that was created. Basically, what it is, it's the first time I think that we have parent-child synchronization mechanism. That means the child put something in its zone and then it means that the parent needs to update their content, their zone with the child's information.

Then, on top of the CDS record, there's two draft that are in progress. One is the DNS operator, RRR model, and I'll cover some of that. This is getting the CDS from the child to the parent and the protocol around enabling that. There's another draft around managing DS record, the maintenance.

With CDS, you can do key rollover. The intent is that all of that is done automatically so that a registrant is enough to copy and paste cryptographic material to a registrar to enable all of this. We want to automate as much as possible with this.

Next.

I'll try to summarize it in a easier way. A CDS is a signal to the parent. A CDS is there to instruct the parent to do stuff. The stuff is you can create the initial bootstrap that means sign the domain for the first time. If you publish the CDS, it means I want to be signed.

You can use that to a signal addition or removal of DS record. Depending on any CDS you have in your zone and how you sign it, you can instruct the parent to do some transaction there. Then if you have a null CDS, it means I want to be unsigned. I want to remove the secure delegation. That's what the RFC and the draft are all made to enable this DNSSEC automation [out of that].

Next one.

Here, the idea is that I'll try to cover how to do a DS key rollover here using CDS and DS. Today, to do a key rollover manually, you need to sign your zone. When you sign your zone, you have to give the registrar your DS record. Eventually, if you do a key rollover, you got to put a new key in your zone, you sign that. Then you need to manually get your DS record to the registrar to the one interface and then you wait a little bit. Then you delete your old key. Then you need to go to the registrar, find out which key you deleted, click on that, delete the key. Then you're done.

You have to go to the registrar's webs interface three times to do a key rollover manually copying cryptomaterial. That's the process we have today which is a standard process. Let's automate all of that using CDS.

Next.

The idea here is you sign your zone and then inside your zone, you publish a CDS. That's essentially, it's a DS record that matches your DNS key. As soon as the parent sees there's a CDS and knows we need to bring that CDS to the parent and create a DS record. CDS means put DS in zone in the parent.

Next.

If you want to do a key rollover, you have your CDS for your new key to sign your zone. Basically, you need to put that CDS at the parent. You publish the CDS. It means the parent sees that, and they need to add that new DS record in that zone.

Next one.

The parent grabs that, the zone. You can wait some time. Then after that, you need to delete or remove the old key.

Next one.

You remove the old CDS inside the child. That instructs a parent to remove the CDS on their site.

Next.

Then you sign your zone. Nobody has to copy, paste DS record from web interface. It's the CDS, what you do with CDS, instructs the parent on its own content. That's easy.

What we did is the draft for the automation, we build the system to automate this. This black box is a piece of software that can run at the registrar. Registrar can run that piece of software or registry can run that. What the software does is based on the instruction that the child puts in their zone with the CDS, the piece of software generates EPP code to add, to create or delete DS record. That's pretty much it.

The reason we have an API in front is that we need to know which domain needs to have a transaction. If I run example .ca and I'm the DNS operator and I signed my zone, I need to tell this software, this piece to go look at my domain, go see if there's a CDS and do whatever the instruction is.

If the domain is not signed and there is a CDS record, that means I need to sign, add the DS in the zone. There's three EPI, three commands that are supported in the RESTful interface. POST is to create the first record. DELETE to unsign, remove the secure delegation and PUT is maintenance activity. It's a piece of software but the only trigger is domain name. This domain do something and that something is defined in the child with the CDS that are present.

We had a lot of feedback in the past around, well, you can't bootstrap a domain just like that and everything. What we did is build extensive validation around it meaning we make sure that the hygiene of the domain that's been worked like example .ca that everything is good. The name server, the DNSSEC is signed properly.

If the domain has three name servers, we'll go to each one of those name server with TCP and get the DS, get the CDS, get the DNS key, validate that everything is correct, that there's no – like if it's a lame delegation, you don't work on that domain. Similar

to what zone master does. Making sure domain has a good hygiene, and then you can do the DNSSEC automation on.

Next slide.

We built a prototype of this. It's live. You can connect right now if you want. It's dsap.ciralabs.ca. You go there. We published the code for the prototype. It's on GitHub at the address there. The other thing we did is we created five-test domain because to play with this, you need to have domains that have CDS, various combination of CDS, so bad, good domain that's in key rollover to add a new DS to remove an old DS so that you can play with different domains. It's all I've been working.

Next.

I'm not going to do the demo. I'll just do the slides. It works. You can go run it if you want. But if you go in and you type “cira”, you do the demo. You connect and do your own demo. There you go.

The reason we built a web interface in front of it is because all it does is that in the backend, it runs the RESTful API, it runs a post or get. But for humans, we have to build a web interface. What we have there is DSAP-1, so go CIRA-DSAP-1, you go Secure Domain. That domain as a CDS for the KSK for that domain, there's a CDS in the zone. Secure Domain means grab the CDS, creates a DS in the zone.

If you're at the Tech Day session, you saw the EPP code that came out of this which is a command to do a create. In here, you can see the DS that's been generated. This would create the chain of trust for the first time for a domain. It would establish the secure delegation for the first time using a CDS.

Next one.

If you go dns.ca, that's a test domain from Cloudflare. You can see it's got some issues with name servers. It's a lame delegation. The name servers are very cold, ice cold. They don't match the parent and child. That means we don't do a transaction.

Next one.

To remove a secure delegation, if you go at CIRA-DSAP-3 and then do a query for the CDS, you'll see it's a null CDS record. What that means is remove the secure delegation. The code would go in, generate the EPP command to remove the DS record for that.

That's a maintenance activity. In this case, it removes or adds a new DS for the domain record. The idea here is that piece of software is meant to run at the registrar if they want or at the registry. For.ca, we could run this and enable all of our DNS operators to sign and manage their DNSSECs using this because

none of our registrars support DNSSEC. We only have 100 signed domain so this would help .ca and the .ca registrant to sign their zone.

In the gTLD world, the registrar could use this to enable DNS operator to sign zones and all that. It's an alternative method for doing registry maintenance with DS. Once your domain is signed with DNSSEC, obviously, you trust the content of the child because there's a secure delegation, it's signed, it's trusted. Then the CDS instruct the activity ongoing.

That's it. Give it a try. We need feedback. There's a lot of people that have different issues with this. The more feedback, the more we can address everything. Questions?

MUKUND SIVARAMAN: I have one.

JACQUES LATOUR: Yes.

MUKUND SIVARAMAN: Hello. This is Mukund from ISC. You mentioned that if there is no DS record on the parent side of the delegation and the child has

set up a CDS record, you have a mechanism to pull that CDS record and create a DS record at the parent side.

JACQUES LATOUR: Yes.

MUKUND SIVARAMAN: Okay. Isn't that insecure?

JACQUES LATOUR: Yes.

MUKUND SIVARAMAN: That's okay?

JACQUES LATOUR: But that's in the registry, we have a delegation. That domain has two or more name servers so we trust that information. Over TCP, we reach all the name servers and we grab the CDS over TCP with two or more name servers.

If we don't trust our own registry in this case, then yes, it's not a signed transaction because it can't be signed. The first time, it's our method of doing this. If you don't like it, like I said on slide

number 2, go to your registrar, submit your DS and sign your domain.

ROD RASMUSSEN: Basically, what you're saying is that if somebody has control of the name servers, they can already do bad stuff anyway so this doesn't make things any worse, I suppose. We've said it a different set of way.

JACQUES LATOUR: If a bad actor has a control over domain, they're not going to play with DNSSEC stuff.

JULIE HEDLUND: Just please do say your name when you're speaking. We know who you are but –

[PAUL WOUTERS]: [Pat Wouters]. I am one of 100 domains, signed domains in .ca now [inaudible]. I just tried to add the CDS record to delete my DS record and Open DNSSEC 149 gives me a syntax error value expected if I put in the null record. If I put in in CDS000, I get a parse error, so we need to work on that.

JACQUES LATOUR: The last stuff we need to work on with this for sure. But this is the precursor to a couple of things. There's a new thing in [hopper] eventually, CNS which is the child is going to instruct the name servers on the parent and there's way more political issue around this. This is the easy stuff compared to what's going to happen. We need to fix this for the rest.

WARREN KUMARI: I just want to mention that the original CDS and CDNS key documents actually allowed this and had the very same stuff in it. The working group at the time said, “Yes, that sounds like a bridge too far.” We said, “Great, we'll publish this,” the original CDNS key fully expecting that people would come along and add it back in. I think that often once you publish a draft, people realize that it's not as scary as they thought and then you can more easily build on it. I think this is great, was part of the original intent, we just took it out and it's got added back in.

JACQUES LATOUR: Thanks. Question? Robert.

ROBERT MARTIN-LEGENE: Robert Martin-Legene from PCH. So this system that you have would allow anyone to contact the registrar directly and the

registrar would have to tell the registry, right? But is that a registrar that doesn't understand DNSSEC?

JACQUES LATOUR: DNS Operator, this runs either at the registry or the registrar. One, two, one, two. And the RESTful API. Essentially, once you have a CDS in your zone, that means I want my DS at the parent. It doesn't matter who initiates the API call because when you have a CDS, that's authoritative to say, "I want to be whatever transaction." Whoever runs it or whoever calls the API, it doesn't change integrity of the system because the intent is with the CDS. Does it make sense?

ROBERT MARTIN-LEGENE: Yes. So basically, the only thing the call does is that it says to the registry, "Please, check my CDS now" instead of automatically you scanning every domain.

JACQUES LATOUR: Yes. So potentially, we could have a [crun] every day. Once a day, we scan two million domain and we do whatever DNSSEC transaction once a day. Once we have two million signed domains in .ca, we'll do that. We'll scan the whole thing every day. Once we have two million. Yes.

JAROMIR TALIR: I have a question. Right now, your registrars don't support DNSSEC but it might happen in the future. Trust me. And what you will do at the moment when – do you expect some collisions like you will still be running these service even when the registrars will be running the same service and somebody will… There are two ways how to change the DS records, directly and via registrar. Do you see this as an issue and if somebody will change DS record directly, will you inform the registrar that some things has happened during the transaction that they may have their own records in the registrar database and they have to also synchronize the data?

JACQUES LATOUR: Yes. We're thinking of doing about two things. One is a poll back for the DS. That's been modified. Then the other option is a registrar lock. That means if one of our registrar did decide to do this, then if we run our own, we can do everything expect for that specific registrar.

[PAUL WOUTERS]: I can add to that. Recently, there's been a new added EPP extension that allows the registry to send the message back to the registrar. They can actually then send an update message to

the registrar saying, “By the way, the DS record got updated without you so here's the updated information.” Then they can update on local information to match it.

JACQUES LATOUR: I have the right to remain silent. That's it? Questions? No more? Lots? I'll be outside. Thank you.

JULIE HEDLUND: Thank you very much, Jacques. Our next speaker is Wes Hardaker. Come on up.

WES HARDAKER: I can use one of these. That's fine. That’s something that works. Yup. There it goes. I hear audio. I hear audio. The camera won't see me. That's true. That's up to you, guys. Do you want me to sit there?

UNIDENTIFIED MALE: [inaudible]

WES HARDAKER: Okay. Yes. No, I'll go up there. That's cool, because I was going to do the quiz from up here anyway because it's much more

dynamic. I do understand that it's 12:00 and I'm the last thing standing between you and the great DNSSEC quiz. We'll try and hurry up.

My name is Wes Hardaker and I work for the University of Southern California at the ISI Department. I'm going to talk today on a project that we are undertaking. It's a rather large project. We're looking for feedback. As we go through at the end, I'm going to ask you for feedback, to e-mail it to me, send it to me, meet me in the hallway, whatever, because we're doing some stuff that I think hopefully might interest everybody.

Next slide, please.

If you look at the evolution of the DNS system today, it was created in 1985, a long time ago and it was very academic. Back then, there was a lot of academic involvement. Then 1995 was the beginning of the huge boom where the commercialization effort took off. In ’98, ICANN was created. In 2004, new gTLDs were introduced. Then finally, we just passed the NTIA transition.

If you look at this timeline, you'll find that we've gone from a fairly academic environment to a fairly commercialized environment where the DNS really has to be quite stable. That makes it very hard to do some experimentation.

Next slide, please.

There's this role change that's happened. We've gone from academic to commercialization. One of the things that's happened is because it's hard to experiment is that starting in 2005, academia has really lacked perspective to be able to contribute new directions and new thoughts and new ideas.

Can we benefit from this complementary role though? Can we take the commercialization side, take the heavy used side and then still actually get some new research projects, some new forward thinking out of it?

Next slide, please.

We've done some things. We've certainly produced DNSSEC. There's a whole bunch of other protocol changes that people are considering though. NSEC5 is out there. DANE is up and coming. There's a lot of tests that we could do around code innovation. The code suites today are fairly static. There's not a whole lot of new ones. Some have popped up more recently. Unbound is newer than BIND and not as newer than Unbound.

Then there's a lot of hardware changes that people could do. Accelerated hardware, for example, would be one thing that we could study further.

But safe experimentation on any of those is really challenging. You can mirror and you can get live feeds and stuff like that and there's privacy concerns associated with it all. This is the type of stuff we want to battle in order to actually do some experimentation yet with real life networks.

Next.

What we've decided is we're going to marry a testbed with an operational network. So at USC we run one of the root servers. We run B-root. We are hoping to marry it along with the testbed so that you can do some tests on real world traffic at the same time, if you want to bring your own zones to play with, or your own protocols to play with, all that should be possible. I'll show you a diagram here in a minute that's more explanatory about what the testbed parallelism might look like.

We want to create some hardware for conducting experiments as well as software for collecting research traffic and analyzing traffic. We want to do some comparisons. What happens if you insert this new technology answering DNS queries? Can you compare the output of the operational network to your new code? Is it faster? Is it slower? Is it producing incorrect results? Is it producing the same data? Those types of comparisons is what we consider one of the keys to what we're developing.

Then, of course, there's a lot of software out there for replaying and testing and things like that. But in terms of a scientific approach, even if we're looking at live, possibly anonymized traffic, you might want to rewind and replay. Your code base didn't quite work well and you want to run it through the exact same things you can get an improvement and then later switch back to maybe live traffic.

Next, please.

This is diagrams that depict our current high level architectural plans. This is where I really want to feedback of if you have research plans or if you have thoughts and desires for things that our testbed might be able to help you with and you see there's pieces missing or you see you have requirements that we haven't thought about yet, this is the type of stuff that I definitely want a feedback on.

Next.

Typically, DNS services are actually quite minimal. You have the line coming in on the left. It hits the firewall which I believe with FW on the first block and then some production boxes that actually handle the service. The larger scale your network is, the more production boxes you're actually going to have fielding in.

Regardless of whether it's Unicast or Anycast, there's probably multiple boxes.

Next.

We wanted to add a parallel testbed. The orange boxes up there are new ones. There's another firewall put in place because you may want to filter production traffic differently than test traffics. We actually have two in mind. Then a splitter where the production traffic goes up and it still gets answered. It still goes out. Then it gets mirrored into the testbed down below.

Down here on the bottom, we have extra blocks. Researchers can put code in some of these blocks. Then there's some blocks in here which would be provided like an anonymizer to anonymize traffic in order to maintain privacy.

But you could have a couple of things. If you wanted to send everything over TCP and see could a system handle real live traffic at TCP levels cannot be done. There would be a conversion ability here. Or I know DNS server HTTP is being considered right now. You could do that conversion there. Then send it through what would be the production equivalent of that name server or anything else you might want to put here.

We have test machines that we expect to be able to log in and use as well as if you want to bring your own hardware, we'll

hopefully have a place for that at some point as well. Then out the backend, you can do the reverse decode if you want to say do a comparison which I think is on the next slide, so let's go on. Yes.

When this comes out, you'd want to be able to compare it against what really happened in the production network. We have a comparison and verification engine that we hope to – if you convert it to HTTP and then it gets sent to through [Jason] over here and then it comes back out, you can compare it to the real data and see if you answered it the same way. That allows for a significantly more verifiable research for new protocols and really what's available today.

Then everything is able to be researched. There's a research archive that will capture data from both the production network and from the testbed network so that you can take your data home with you. Once you're done with your experiment, you get to walk away with it.

Next, please.

Also, we have some traffic generation ability actually already. A lot of the stuff is very new. We don't have most of these blocks in place yet but at ISI, we actually have some technologies that we've actually developed ahead of the rest of this. There's a

traffic replayer that we have locally which is actually quite good to be able to play it line reads from pick up files and stuff.

Next.

Then, finally, we have this insane theory that we could even, at some point, once we were absolutely sure that this technology down here is verifiably perfect, we can actually begin letting in enter real traffic out to the real world again. That piece is subject to much debate at this point. There's no reason why we think it would be a necessarily bad thing if it can be with operationally 100% safety. There's a big red stop button that goes along with it.

Next.

Then, of course, one research testbed isn't good enough. We wanted to have three running in parallel so that multiple people can come in and do these experiments at once. This is not something we're going to have tomorrow. Again, this is our long-term vision. We'll probably start with one and we'll build up more as more people find it useful and popular.

Next, please.

What can we do? Here's some example configurations based on the diagrams you saw. You can run in parallel. As I mentioned,

you can replay and generate traffic. We expect users and zones to be able to hosted too so that we could actually offer a production set of boxes that would actually run your real zones and then you could run parallel stuff with your own data. B-root is just one of the zones that we have available to play with but there's no reason that somebody could bring their own or at least a portion of their own if they want to service an Anycast address at our facility.

Protocol conversions and testings, I already gave some examples to that. There's a lot of that being considered right now. There's box coming up the next IETF on that very topic, in fact.

Address based operational and test separation of what else can you do. In that entire architecture, we're very much looking for feedback on what types of things you think might be beneficial? What research ideas are coming up from the top of your head is.

That would be really cool if I could do this. We want to hear that so that we can use that when we're pitching the eventual creation of this to other people.

Next, please.

This is the run in parallel, user or root zones. I talked about this already. The testbed runs in parallel.

Next.

If we're going to do a replay and generation, I talked before, one of the nice things is because everything is archived. If you decide you want to try that again because it didn't work, we can actually rewind a couple hours or rewind a couple of days or weeks and say, "Okay, we have modified our code, we can take that research archive and put it back into the traffic replayer and replay it again and see if you've improved your code."

Next.

Then finally, we've talked about this slide already as well where possibly on a perfectly validatable solution that it appears like it's looking perfect we can actually let real world traffic go out too. You can switch over gradually from the production code that you all know and run and have been for decades to brand new stuff that nobody has ever seen before and see if the rest of the world notices. They shouldn't because we should have already verified it with a comparison engine.

Next.

Our goal is safe experimentation as my colleague likes to put it. You want to test new tires but you want to test new tires on a running car. That's hard to do. But critical infrastructure cannot fail. If your zone is critical or our zone is certainly critical, it

simply cannot fail. But research means trying new things. We're trying to figure out a way to marry those two concepts and actually allow you to do new things but not let it fail. We have new infrastructure for doing all that kind of stuff.

Next.

We also want to create a community. We're actually going to host some workshops. There's actually going to a workshop hosted next week in LA on this subject. We want to outreach to academia. I'm going to hold a [buff] at the IETF next Monday night. If anybody is interested, please see me afterwards and if you're going to be in Seoul.

Then we're going to outreach the operational communities. We're going to bridge the gap between academia and operational communities again. As I said, looking for feedback if I haven't mentioned that enough yet.

The best thing is that all of our tools would be open source. If you want to go build this at your own facility, we'll release our diagrams and code and everything. We'll have a place for you to go and play with if you want. But if you want to go build it on your own because you think it's that useful, you want to run it 24/7 all the time and not subject to our kicking you out after a

month or whatever we decide the operational period should be, everything will be open source.

Next.

The overall benefits are we're really trying to figure out how we can grow the academic involvement again around DNS and do some new research. We want to accelerate innovation. We really want to push what's the boundaries especially as things like the Internet of Things come online and they all need a name. There's a lot changing in the world that I don't think we fully understand or predicting that. If we don't get ahead of it, the Internet of Things is going to be far less useful than it could be if say every device has a name that we could all use.

Again, we want to collaborate with academia industry and governments and nongovernment organizations and bring them together again. Right now we're sort of split. Academia is off doing their own stuff. Operational is off doing their own stuff and government's here.

Next.

As a timeline, this is where we are now. We're gathering requirements which is why I want to hear from you. We're looking for sponsors for actually pulling this off both on the hardware, time and financial and all that kind of level. We're

trying to build a community and find some collaborators. If you want to collaborate with us, again, we want to hear that as well. If you want to actually be a part of this, we'd love partnerships.

Not in too different future, we intend to buy the needed hardware and create the needed software to make all of this possible. Then eventually open it up to researchers for experimentation so that people can come in and play.

Next.

That's it. Join us. We're looking for feedback. That is our old URL because we just came up with a name for this like last week. These slides were made two weeks ago. You can still get it. If you go to that URL, it'll redirect you to the right place but we used to call it the research route but we actually wanted to be much more big and expensive than that.

The new name is NIPET which is Naming and Internet Protocol Experimentation Testbed, so /nipet we'll work to.

We have a mailing list already set up. The mailing looks like it's attached to our workshop that we're holding next week. But the reality is it's mailing list, generic to this whole project. Please, do send us ideas, suggestions and feedback. In these cases, find me afterwards, I'd be happy to give you an e-mail address.

Join the community and attend our workshops. We're holding one next week. I realized none of you are probably going to be able to attend the one next but we hope to hold them on an annual basis for really pushing the DNS technology. Thank you very much. Any questions? It's either that or I'm going to ask you questions about the quiz in a minute. Jaap.

JAAP AKKERHUIS: Yes. I was wondering how this compares to the other experiment known as the [GitHub]. Sorry about that.

WES HARDAKER: No, it's a very good question. YETI is very different architecturally looking, and I'm very familiar with YETI. I could see YETI wanting to come in and say run an instance of YETI in our testbed in order to do that parallel verification. They're not doing that. They don't have real traffic you know they have – it's a set – no real traffic. They have real traffic but it's a separate set of traffic.

I think in the future, I could see us working with YETI but we're trying to be a little bit bigger and larger than what YETI is focused on which is really just DNSSEC quick changes at the root and other various things that you could do. We'll likely collaborate at some point. Warren?

WARREN KUMARI: I might have missed it because I wasn't paying attention the whole time.

WES HARDAKER: You did.

WARREN KUMARI: Do you have anything about the privacy implications and stuff like people have to sign NDAs before they get copies of data? How do you deal with that because [inaudible].

WES HARDAKER: Yes. One of the diagrams that you may have missed actually had an anonymization engine in the front end and afterwards. It is there in terms of the legality for how we're going to handle that, the legal issues and paperwork have not been worked out.

We already have anonymizing technology within ISI. We actually have an entire set of people working on anonymization engine. We already run one in order to do diddle data and other stuff like that. No. We already have that capability. There's always a question of how much you anonymize and what and things like

that. It falls out, legally, I'm not sure yet. If you have requirements, I'd love to hear them though. Ric.

RICHARD LAMB: One of the staff, I love it.

WES HARDAKER: Thank you.

RICHARD LAMB: How do you envision this? The first thing that occurred to me when you gave this presentation was this is great but from a commercial point of view – in the distant past, Harvard had a router testing lab. [inaudible] various academic institutions had a place for vendor. I'm just speaking purely commercial.

WES HARDAKER: Sure. Yes.

RICHARD LAMB: I have a product, I want to test it. I come in here, I pay a certain amount of money, probably. That's probably to be determined. I put my product there and I get not a certification like an FCC but some sort of stamp on my product that says it's been through

some – used the word “verification” a few times. That made my ears perk up. That would be a great thing.

WES HARDAKER: That's certainly very interesting feedback. We are thinking a little bit more academic than that but you're absolutely right that there's nothing that our verification engine couldn't produce a result saying we're 99% compared the same as BIND.

Yes, that is possible if commercial players actually had real world to be deployed tomorrow kind of stuff that they wanted to bring in, I don’t think we’d be against that. In terms of a fee, ideally from a purely academic perspective, we want this to be as open as possible. We're not trying to charge fees for especially other universities and stuff coming in and playing our true research. We don’t want to. But we haven’t yet figured out the entire sponsorship roles and everything.

[DAVEY SONG]: Hello. This is Davey from [BI]. Actually, Jaap asked the question about the difference between YETI and your testbed, and I would like to make some adding that your slice, I didn’t catch up some research engine that changed the UDP to TCP and HTTP. Is there any other research items already have in your agenda?

WES HARDAKER: We don't have a fixed list of stuff. We have some experiments that we know we want to do. A DNS privacy is actually one that my colleague John Heidemann at ISI is experimenting heavily with. That's one of your immediate targets of what we play with. No, we expect that framework to be able to handle anything that you might want to do conversion wise, algorithm conversions and DNSSEC. There's lots of stuff that could happen. I'm not trying to give you specific ideas because I want you to bring ideas and thoughts.

[DAVEY SONG]: Yes. I think currently YETI are running for more than one year and we connect some thoughts and some effort. I do hope that the people who want do some new things can work together, can think together put some pieces together.

By the way, I noticed that your testbed have currently the picture, the high level design is focused on the assertive side and not reflect the… The resolver behavior is very important to design the assertive side. Currently, what YETI focus on one part is on the structure how root zone or any zone can be signed, can be produced. Secondly, how the information can be distributed

to the end users, the resolvers. That's what we are focused on. I do hope that we can talk about that some other [inaudible].

WES HARDAKER: Yes. I think that's a wonderful idea. I think that the architectural framework would fit testing resolvers quite well but you've got to get enough clients to point it a resolver and that's the tricky side. No, that's a great point. I'll help to make sure that we think about that further.

[DAVEY SONG]: Yes. Next week, we will meet in IETF and also we have a workshop one day before the IETF meeting on September afternoon.

WES HARDAKER: Yes, I saw that.

[DAVEY SONG]: Yes, we will meet soon.

WES HARDAKER: Okay. Great. Thank you. I think there was one more question then we're probably out of time.

UNIDENTIFIED MALE: Can you hear me?

WES HARDAKER: Yes.

UNIDENTIFIED MALE: How many testbeds you want to have established all over the world or only in U.S. you won't to have it. If you are entrusted in India, we can plan it up and we are working on project [inaudible] million people in especially in Pradesh, in South India, to bring the people on to the [inaudible]. We want to know more how we want to see.

WES HARDAKER: Right. How many testbeds we want? That's a wonderful question. We want as many as are needed in order to keep people happy. We started a project at ISI 15 years ago. I wasn't there at the time but I actually helped create the report. They started the whole project 15 years ago and then they took it and ran with it. It's a project called DETER which is a very elaborate testbed. That has been growing ever since and it is a huge project within ISI.

How many we want and are we going to host them in multiple locations is yet to be determined based on how much feedback I get and how much need and how much other people say, “If this existed, yes, we would love to make use of it. Please keep me informed.”

That's a good question. But I don’t have a definitive answer because it depends on interests from the community at large like you all for example.

UNIDENTIFIED MALE: How much is the funding you plan? How much budget you have made on this one? I do remember [inaudible] Internet to project [inaudible]. I don’t want to be like that the same. What is the budget and how you want to collaborate, you let us know. We can think about it actually.

WES HARDAKER: Okay. Yes, no. Please contact me afterwards. I'll make sure I'll give you my business card so we can keep in touch. All right. Thank you very much. All right. We want to switch. I'm still too on deck.

All right. Our usual quiz runner is not around so I volunteered to do it this time. You, hopefully, all have a piece of paper. You'll be

happy to know these questions are all letter perfect. The answers are perfect. I know because I wrote them at 3:30 this morning. I'm quite sure they're accurate. My answers always win. That's the clause that Roy always uses so I'm going to use it too. If you all disagree with me, I'm still right. That's the way the game works.

JULIE HEDLUND: Look around for the answer sheets. There should be answer sheets. They actually have space for ten questions but we have eight. That's because we were anticipating a different quiz. But instead, we have a better quiz.

WES HARDAKER: Thank you. Yes, Warren?

WARREN KUMARI: [inaudible].

WES HARDAKER: There are a couple of questions which allow you to answer more than one per choice. It should be fairly obvious but if you have a question at that one in a particular time, do let me know. You'll be glad to know that I hate history questions so there are none.

UNIDENTIFIED MALE: What are the stakes?

WES HARDAKER: What are the stakes? The stakes are name recognition. At the end, we will count up the number of points. There's eights questions but there's actually a total of 29 points which give you somewhat of an answer for where we're headed. It shows that you are the most DNSSEC savvy person in the room. That is a big title, my friend.

All right. Question number one is how many hours are left in the TTL for the U.S. election? No, just kidding. I was really tired of listening to CNN this morning.

I thought we'd start with an easy one. What are the first three fields, in order, for the RRSIG wire format? How many people know this one? The choices are A) type covered, algorithm and then the original TTL value. B) type covered, key tag and then algorithm. C) type covered, signature expiration and then signature inception. Or D) type covered, algorithm and labels.

Okay. When you know the answer, put it down on your sheet under line 1. If you don't know the answer, you might guess

randomly. I will give you a hint because I'm a helpful guy. It starts with type covered.

Next. Number 2, a little bit easier honestly. What does DANE stand for? If you've read any of the DANE documents recently, it'll be easier. A) DNS Based Authentication of Named Entities. B) DNSSEC Authenticated Named Entities. C) DNSSEC Authentication of Naming Entities. Or D) DNS Based Authentication of Naming Entities. Again, I'm a helpful guy, it ends in entities. Pick one of those.

Next. Number 3. I lied. There's a history question. Number 3, in what year was the DS record for .com put into the root zone? 2010 BC, the year of [Haguenau]. B) 2011 AD. C) 2012 AD. Or D) 2013 AD. I'll give you a hint. It's not A.

Next. Number 4, which of the following terms are defined in RFC4033? For those of you who don't know, 4033 is the overview document of DNSSEC and how it works. There's a big terminology section.

A, so this is one, Warren, that is multiple choice. Here's the rules for multiple choice ones. There may be more than one right answer. There may be zero. If you put down multiple answers, as long as they're all correct, you get a point per answer. If any of

them are not correct, you get zero. You may only put down the ones that you're sure of. That would be my suggestion.

A) a non-validating security aware stub resolver. B) a security aware recursive server. C) a validating stub resolver. D) a security aware resolver. Or E) non-validating stub resolver. I'll give you second to consider those. You can put down up to five letters. You can put down zero but I guarantee you'll get zero points for that because at least one of those is right.

All right. Next. How many TLDs in the root have their zone contents signed? This should be easy because it was on the slides I presented this morning. You've already seen the answer, all you have to do is pick the right one.

A) 3.1415629. B) 1279. C) 1349. Or D) 1509. I'm going to give you two hints. It's not A and it ends in 9.

WARREN KUMARI: Can I put down multiple answers?

WES HARDAKER: If you want to put down multiple answers for how many TLDs are in the… Yes, go for it, Warren.

Next. Number 6. What is DURZ stand for? Remember, when the root was signed originally, there was the slow rollout mechanism and it was called DURZ. What does it stand for? Is it DNSSEC Unverifiable Root Zone? Or B) Deliberately Unvalidatable Root Zone. Or C) DNSSEC Upcoming Root Zone. Or D) Deeply Urgent Rabbit Zebras. I'll give you a hint. It ends in zone.

Number 7. How many RFC up to RFC8009, because that's how many there were at 3:30 this morning, contain the string DNSSEC? In any case variant, uppercase, lowercase. There's at least one I saw that has DNS in uppercase and then sec in lowercase. The answers are A) 3. B) 58. C) 142. Or D) 275.

UNIDENTIFIED MALE: You mean the title or the whole –

WES HARDAKER: No, the whole thing. For those Unix geeks, I did a graph of the entire RFC set of documents this morning. Those are the numbers of RFCs that came up. It's either 358, 142 or 275. Yes, sir?

UNIDENTIFIED MALE: [inaudible] we can look at?

WES HARDAKER: Later, I'd be happy to send it to you. Yes. All right. Next, 8. Here's where you can earn lots of bonus points. Now, this one's about the title. The last one was anywhere in the document. This one is write down any RFC numbers up to, again, 8009. There could have been another one published since then. I don't know.

Any RFC numbers that with a title that contains the word DNSSEC. I guess I don't know if that's a word or not. We'll call it an acronym but it's sort of an acronym. The phrase DNSSEC you will get one point for every correct number but negative 5 if you write down an RFC that does not have a DNSSEC in the title.

This is where the big points come in. This is where if you can get them all, you can get up to 29 points total in the quiz. I'll give you a minute to think about that one because it's certainly not easy and it will test your knowledge of the IETF quite well. Or I can give you a hint. There's 19 of them.

When you are done, the proper way to play this game, I've been told many times in the past, is to exchange papers with your neighbor so that your neighbor gets to grade your paper.

WARREN KUMARI: Wes, can I [inaudible]?

WES HARDAKER: Sure. No. You can guess random letters, Warren. I really suspect you should guess random letters for number 8. That would be cool. All right. If you exchange papers with your neighbor or promise utmost honesty but it's a lot more fun to laugh at your neighbor while he laughs at you or she. Excuse me.

All right. Next slide. Signature expiration, times up. What are the first three fields in order for the RRSIG wire format? By the way, I would –

UNIDENTIFIED MALE: Can I make a small comment?

WES HARDAKER: Please.

UNIDENTIFIED MALE: I think the first part of the wire format is the label and not the R data part.

WES HARDAKER: The first format in the wire format according to the RFC is type covered. That is the first bytes in the RRSIG field.

UNIDENTIFIED MALE: Of the R data part of the RRSIG.

WES HARDAKER: True.

UNIDENTIFIED MALE: You didn't say R data part, you said Y format.

WES HARDAKER: Then you should have put no letters down. But again, at 3:30 in the morning, this was truly correct in my mind. I'm going to stick with it. I woke far too early. Type covered, algorithm and labels. I could not have answered this question to make you feel better if you did get it wrong.

Next question, what does DANE stand for? DANE stands for DNS Based Authentication of Named Entities. Number A. I woke up far too early.

Next. In what year was the DS record for .com put in the zone? It was 2011. That is when the DS record was put into the zone. That, by the way, was also on the slide from this morning. If you weren't here this morning, you lose the game. See how that

works? You're supposed to show up promptly at 9:00 on these events.

UNIDENTIFIED MALE: Gregorian calendar.

WES HARDAKER: Yes, on the Gregorian calendar. That's a good point, which is the only one I use at 3:30 in the morning.

Okay, next. Which of the following terms are defined in RFC4033? All of them pretty much. I just removed one word from one of them. The rest of them, A, C, D and E were all defined in 4033, and a whole bunch more. The number variations of terminology that they combine that they decided needed to be defined was quite large. Security aware recursive name server was defined but I removed the word “name”. Yes, Ric?

RICHARD LAMB: [inaudible].

WES HARDAKER: That's zero. That was a zero. According to the rules that I proclaimed when this slide appeared, you get zero. If you put B, you get zero. If you put any other letters, you get one point per

letter. Isn't that easy to score? You just count one, two, three, zero.

All right. Next. How many TLDs in the root right now have their zone content signed? Which, by the way, at 3:30 in the morning, I was reading off the slides from this morning and I realized that the slides we're horribly worded because if you read it the way are slides are written for the introductions, it actually read 1 because it would have been the root. I had to reword that sentence very carefully to talk about content of the lower ones. 1349 TLDs have their zone signed.

UNIDENTIFIED MALE: [inaudible].

UNIDENTIFIED MALE: Yes.

WES HARDAKER: Ooh. Good thinking. Okay. I will give credit for B or C. I'm a nice generous guy, so anybody that put B or –

UNIDENTIFIED MALE: [inaudible]

WES HARDAKER: No, because you can't have 3.14 zones. Nice try. If you put A, let's talk afterwards. Next.

UNIDENTIFIED MALE: We need to talk if your counting is 1350.

WES HARDAKER: Then you should have made the slides for this morning. You should talk to Dan York because that was actually from him. Okay.

Six, what does DURZ stand for? It stands for Deliberately Unvalidatable Root Zone, B. Not zebras.

Next. How many RFCs contain the string DNSSEC? 275. My original plan for number 8, I was going to have you list any of these. Then when I got 275, there was no way I could put all those on the screen to have you verify them. Yes, 275 RFCs. The very first one is actually like 1000 and something. It's actually the DNSSEC working group is listed in reference to a document that had nothing to do with DNSSEC.

Next. Write down any RFC numbers that up to 8009 that contains DNSSEC. You get one point again for every correct number and

minus five points for every incorrect number written. That's the list. Is everybody able to read it in the back?

You can't get minus one. You get minus five if you put one down that's not on there.

UNIDENTIFIED MALE: It's overall scores.

WES HARDAKER: It's overall scores minus one. All right. I don't usually remember my own RFC numbers, Warren, so I totally get that. All right. Once you're done grading, pass your papers back to your collaborators and we will then do a measure and see how well everybody did. When you know your score –

UNIDENTIFIED MALE: You aren't Roy.

WES HARDAKER: You aren't Roy. No, you don't.

UNIDENTIFIED FEMALE: [inaudible].

WES HARDAKER: Thank you, Paul, because I wasn't going to ask that question. Who has less than zero points? Yes. Okay. Let's go negative first. Who has less than five points, negative five points? Less than negative ten points? You win.

Okay. Now, try to find the other boundary. Okay. Who has more than five points? Whoa. You have five? You have six? How many do you have in the back?

UNIDENTIFIED MALE: Seven.

WES HARDAKER: Seven. My good sir, you are the winner for the day. Congratulations. I'm the winner too because I created quiz that had less number of point than any that Roy has ever created. Thank you very much. I think Julie will now talk about lunch. Right?

JULIE HEDLUND: Yes, indeed. Lunch, as you can probably guess, is not in the room. It is in La Cantina Restaurant. That is in the Novotel. It's

not very far away. If you go out of here and you go to the right towards the Novo –

[END OF TRANSCRIPTION]