SAN JUAN – DNSSEC Workshop, Part 2
Wednesday, March 14, 2018 – 10:30 to 12:00 AST
ICANN61 | San Juan, Puerto Rico

RUSS MUNDY: I'm not sure if Geoff is actually going to be part of the presentation, but Geoff is one of the co-authors, and he's here with us. We have a 30-minute session now for the KSK Sentinel presentation.

WARREN KUMARI: I can sit. I'm fine. I'm lazy. I'll sit.

RUSS MUNDY: Okay. Warren, you're up.

WARREN KUMARI: I'm just trying to figure out where I can sit and still reach a mic. There we go.

RUSS MUNDY: I'll point the timer at you.

WARREN KUMARI: Thank you. Hello, everyone. Looking around the room, I'm guessing a number of people have already seen this presentation.

UNIDENTIFIED MALE: Only 12 times though.

WARREN KUMARI: Is there anyone who hasn't seen the presentation?

UNIDENTIFIED SPEAKERS: [inaudible]

WARREN KUMARI: Okay. Well, I guess this'll be much more interactive then. Basically, this is a joint work with myself and Geoff Huston, who is over there, and Joao, who I don't think is in the room.

Oh dear. Geoff's coming over to help.

So what is the problem we're hoping to address with this? Well, as a few people have mentioned recently, we want to roll the DNSSEC trust anchor, the DNSSEC KSK sometimes. Users who do not yet have the KSK or users who are using validating resolvers which don't have the new KSK will break. Basically, all of their DNS will stop. This basically means for them that their

Internet will stop because without the DNS, nothing really works.

More worryingly, we don't have any way of measuring the actual deployment of the new key. This means that we don't know how many people or who will break when this happens.

We heard recently that there's this thing, RFC 8145. This was published in, I think, late – actually, it will say here – in mid-2017. What this is supposed is it signals which trust anchors specific resolvers have. This sounds pretty much exactly what we want. We'd like to know what resolvers have what keys. This will tell us how the key roll is going to go. This will let us know if it's safe or not.

Unfortunately, no. What this does, as I said, is it provides reporting from DNSSEC-validating resolvers. It lets DNSSEC-validating resolvers tell you what particular keys they have.

In my basement, I have a DNSSEC-validating resolver. It runs in a Docker container, which means that I restart it – actually, let me unplug. For some reason, I'm the translator is translating me to something else. There we go. That helps.

So I have a validating resolver in my basement. I restart it every day, for various long and complex reasons. This means that the validating resolver in my basement never gets a chance to learn

the new key. In order to learn the new key and start trusting it, you need to [see it], and then you need to have it live for 30 days. My resolver never has that because I restart it and it starts up clean.

However, I don't think this is actually important. Nobody is actually sending it any queries. It's a resolver that falls in the forest. It doesn't make any noise because nobody is actually using it. But it's reporting statistics.

This means that it shows up in the graph of resolvers and resolvers that are doing RFC 8145 reporting. For those who haven't seen this graph before, this is a graph that was published by Verisign in, I think, October 2017. It shows the progression of the new trust anchor being deployed. In mid-July, the new KSK was published. In mid-August, the hold-down timer expires, which means that everybody should start learning the key and should start putting it in their trust anchors and should start reporting that.

For some reason, only most of them did. But around 5-7% of resolvers, for some reason, did not pick up the new key. If people were watching the presentations earlier today, they saw that the number has jumped to 30%. We're not entirely sure why.

But whatever the case, RFC 8145 only provides reporting from resolvers and does not provide reporting from users. What we

think is much more important is an ability to understand what the user impact of the KSK roll will be. It's all fine and good to know that the set of resolvers is going to be okay or not, but we don't really care about that. What we care about is that, when the key rolls, will users still be able to use the Internet?

We have a new process. It's called KSK Sentinel. It requires a very simple set of updates to resolver software. It allows anybody who can publish a name in the DNS to set up a measurement service.

But more importantly than all of that, it exposes the result of the tests to the users. The current measurement system provides reporting from the resolver to the root server operators and the IANA – also the IANA through the root server operators. What this does is it lets individual users have a look at their resolver and understand if they will be okay or not.

It also allows for large-scale Internet measurement. I'm guessing most people here have seen some of the presentations which Geoff has done, where he serves Google Ads or similar to a huge set of the population and then gets millions of people to run a test. The plan is that Geoff will do a test using this infrastructure, the KSK Sentinel, to actually sample millions of users and get much better reporting on how they will work.

How does this KSK Sentinel thing actually work? The changes are two very simple rules. A validating resolver that has been upgraded to support this will do all of its normal DNS validation process, all of its normal DNS resolution. When it gets a question, it goes off and does its standard set of work. As the very last thing that it does before sending back the response, it has a look and it sees if the leftmost label in the queue name in this question that was asked starts with this magic string. The magic string is kskroll-sentinel-is-ta-[key].

If it sees this string and if it has that particular key – the one that has that key ID – then it replies normally. If it does not have that key ID, then it takes the valid, correct answer and it returns a SERVFAIL. Basically what it does is it pretends that, while it was doing the validation, it had some sort of issue validating the answer, or while it was trying to do the resolution, it ran into some sort of answer. SERVFAIL is now the generic something-went-wrong error.

The second rule is basically the inverse of the first. It's ksk-sentinel-not-ta. What this question is asking is, "If you do not have the key, then you should reply normally. If you do have the key, then you should turn that into a SERVFAIL.

When you first look at this, it seems a little weird that we're asking two questions, one being the inverse the other. These

can't both me true at the same time. Yeah, that's right. That's exactly the point. These can't both be true at the same time.

To see why, here's a little example. I am a DNSSEC-validating resolver. I support Sentinel. I realize I don't really look like a DNSSEC-validating resolver – they're normally black and square and I'm not – but I am a DNSSEC-validating resolver and I have the new KSK. The key ID for the new KSK which is just being released is 20326.

I'm walking along and I get a query for invalid.example.com. This particular name, as it implies, is an invalidly-signed record. I do my normal DNSSEC processing because it's invalidly signed. I send back SERVFAIL. Nothing has changed yet.

I then get a query for kskroll-sentinel-is-ta-20326.example.com. I do all of my normal work. I do the normal resolution step. I do the normal validation step, and I get back and IP address of 19202.23. Now, this is just an example. I get back whatever the zone publisher has put in the zone.

Because I support Sentinel and because I'm using key ID 20326, I just answer normally. The question asks, "Is this a trust anchor that you have?" I do in fact have that trust anchor, so I just forward off the answer that I got.

A few seconds later, I get a query for kskroll-sentinel-not-ta-20326. I do all of my normal validation process. I get back whatever IP address I get back from the authoritative server.

However, I do have and am using ID 20326. Basically, I do not not have this trust anchor. So is send back the SERVFAIL. Basically, The Simpsons: "I'm not not licking toads." I do not not have that TA.

[RUSS MUNDY]: Warren, could you mention the slide number when you change for the remote participants?

WARREN KUMARI: Sure. For the remote people, it is Slide 7. So all we've done so far is we've added a whole bunch of complexity to the DNS. Adding complexity to the DNS is all fun and good. Is it actually useful? If so, how?

I think that most people can see the slides up on the screen: invalid.example.com/fish, kskroll-sentinel-is-ta-20326.example.com/kitten, and kskroll-sentinel-not-ta-20326.example.com/puppy.

The plan is we ask users to please go to a webpage that has all of this on it. Then we ask users, "When you [are in] this webpage,

do you see a picture of a fish? If you see a picture of a fish – you'll also see a kitten and a puppy – then that means you were able to successfully resolve the invalid.example.com domain name. If you were able to successfully resolve invalid.example.com, that means that you're not using a validating resolver. If you're not using a validating resolver, you really don't care about the KSK roll. In fact, you don't care about the key at all. It in no way affects you.

If you're able to only see a picture of a kitten and a puppy, then that means you could not get the invalidly-signed record – using a validating resolver – but it means you were able to get both the is-ta-20326 and the not-ta-20326 names.

Seeing as you can't both simultaneously have and not have the trust anchor with that key ID, that means that the resolver you're using has not yet been upgraded to support Sentinel. This means that we can't tell you if you will survive the key roll or not.

If you only see a picture of a kitten, then that means you were able to load is-ta-20326 and you were not able to load not-ta-20326. That means that you will be perfectly fine during the key roll. You have the new key. It's in your trust anchor store. You will be just fine.

If you only see a picture of a puppy, then that means that you only have the old key. You do not have 20326. This means that,

when the KSK roll finishes and the existing key that you're relying on is removed, your DNS will break.

Are we really planning on users using pictures of kittens and puppies and fish? I wish we were. It'd be really awesome. Unfortunately, no. It doesn't really scale to ask every user on the Internet to please go off and tell us what one of the pictures of kittens and puppies and fish they see.

Instead, we will serve them a big blob of Java script. What it does is it tries to load the kitten and the puppy and the fish equivalently. Then the Java script looks to have a look to see which of these was actually able to resolve. From that, it runs the test and tells you how you will be.

Yeah, I know it would be awesome if it was kittens. Unfortunately, still no.

We have a demo of this at www.ksktest.net. Because I have a little bit of time and they crazily let me use their computer, let's make sure that it does actually work. So it's www.ksk.net. It is now going off. It's trying to load all of those records. It already did it. It is determined that the resolver that this computer is using is a validating DNSSEC resolver, and it is a legacy DNSSEC-validating resolver. It does not yet support the Sentinel method. That means that this particular machine – and I guess everybody

else at the ICANN meeting – we can't tell them if they will survive the key roll or not.

Here's just a screenshot of that because I wasn't sure if I would have working Internet here.

So that's – sorry, Slide 9 – that's the 50,000-ft. view. I realize that I tempted you with kittens but you did not get any kittens. So here is the obligatory and somewhat gratuitous kitten picture.

Questions? And questions other than, "Why are the kittens doing gardening?" because I don't know.

Frederico? Is that Frederico? Yeah.

FREDERICO NEVES: Hi, Warren. Are you collecting the results of your test page?

WARREN KUMARI: Kind of, maybe. On my particular page – this is just running. I have an Apache web server that serves these resources. So I'm logging the fact that people are asking for the three records, but what's not actually shown is that each one of the records has a random number appended to it. And they're different random numbers.

So, yes, they're being logged. I'm not looking at the logs. I don't have enough information from the logs to know – oh, I guess I do

have enough information from the logs to be able to tell what the percentages are. But, no, I haven't bothered looking. To be honest, I have no real intention to look because, so far, most of the resolvers have not been upgraded. This was purely a proof of concept. If you look at the code, you'll definitely know it was just a proof-of-concept toy implementation.

So, yeah, I guess I do actually have enough info to look. I have no plans to because I haven't through about the privacy implications of that.

Good question, though.

VIKTOR DUKHOVNI: Warren? Hi. Are you aware of any work to get us beyond our RFC 5011? We can measure all the ways in which it's not working, but maybe for the next KSK rollover five or six years from now we can do better. Have you any thoughts about that?

WARREN KUMARI: Well, yes and no. I think that Frederico has a follow-up question, so I'll let him go after this. Wes Hardaker and I have been writing a document pointing out that the 5011 document is vague in some ways. One of the ways that it's vague makes it fairly dangerous and easy to shoot yourself in the foot. But there are a lot of other things in it which probably could be improved.

Much of the problems so far, I think, has been things which aren't necessarily 5011's direct fault – either that people have implemented 5011 incorrectly because they couldn't quite follow the writing style, or, much more likely – these are just my views – people configured the DNSSEC trust anchor as we told them initially, which as, "You should open your BIND or named.conf file and say, "Trust keys equals (put in the key)."

For those who don't run BIND, the trusted keys stanza says, "This is the trust anchor. This will always be the trust anchor. Don't bother trying to roll the trust anchor because I will tell you when you need to replace it."

We were telling people that because that was the best practice at the time. Actually, that was the only practice at the time. After we started telling people that, 5011 support was introduced into resolvers. But that required a different knob to enable it, and we think people didn't enable it. They didn't bother changing trusted keys to managed keys.

The third problem with 5011 or key rollovers in general is that it seems fairly common for people to be using things like VMs or Docker instances now to run their name server. 5011 says, "When you see a new key that's signed with the old key, don't trust it for at least 30 days." This means that, if you have a non-writable file system, like you do in Docker, or if you've made

your server secure by removing BIND's ability or whatever name server's ability to write stuff, you won't ever learn the new key.

So most of those problems aren't actually issues themselves with RFC 5011. They're with users configuring things incorrectly because they followed what we told them to and then didn't hear the new info. Or because the system can't write the new key anywhere anyway. This means that, if we replaced 5011, you would still have that set of problems.

But, yeah, I agree that 5011 has lots of warts. Many people, or at least one person, would say, "Because we expected there to be an operational document and there should always be two KSKs in place at all times. Then it works better." But, yeah, agree. 5011 should be replaced.

There has been a little bit of discussion on replacing it. Much of that involves doing things like pulling the new trust anchor from www.iana.org and trusting it because it's signed with a CA cert. So, yeah, there has been some discussion, but I don't think that they're better ideas.

VIKTOR DUKHOVNI: Right. Just a follow-up comment. I'd like to see the ability to roll forward from any past trust anchor that you're configured with, with whatever security risk you might therefore have, forward

from there through a chain of signatures in DNS, immediately, without a 30-day hold-down.

WARREN KUMARI: Yeah. I guess you could actually kind of currently do that, except you start with the old key, wait 30 days, and then move to the new key and wait 30 days. That would be fairly horrendous.

VIKTOR DUKHOVNI: Yeah. So let's get rid of those 30 days. Make it zero.

WARREN KUMARI: It would also make testing way easier. Actually, I run another site, which is… keyroll.systems? I can't remember. I wonder if it still works. Let's find out.

Look. It's a new gTLD. They are actually in use…. oh, wrong place. Keyroll.systems.

Hey. It still works. So, yeah, this is a quick demo site that I set up a long, long, long time ago. It basically allows people who run 5011 to do testing. It's a key that rolls every 90 minutes.

The problem is that, when I first set this up, the majority of resolvers weren't actually waiting 30 days to install the new trust anchor, so you could just point your machine here it would just work. People had basically ignored the and-wait-30-days hold

time. Then resolver s had it pointed out to them that they weren't following the RFC, so they started following it and waiting 30 days. So getting them to not wait 30 days would kind of be a regression.

Anyway, it's good to know this still works.

Who else had questions? Frederico?

FREDERICO NEVES: Very first follow-up, actually, for all the authors, actually. What do you expect if everything goes [smoothly] now in the following weeks in the IETF? This is mostly for Geoff, I think. Would you wait until we start to do your tests? Will you wait for a month and hope that everybody deploys those new resolvers? What do you expect?

GEOFF HUSTON: Necessarily, this is a change to the behavior of resolvers that are operating in a mode that validates the answers that they send back using DNSSEC because every answer now needs to look at the leftmost label, figure out if that's a key piece of text, and, if it is, invoke this behavior.

At the moment, to my direct knowledge, a module in the Knot resolver from CZ.NIC has been implemented this, but BIND,

Unbound, and all the other stuff folk use have not implemented it.

So I suppose, if I really wanted to find the market share of CZ.NIC's Knot resolver, I could unleash a test right now and it would show me that, but that, while useful to CZ.NIC, wouldn't be useful to anyone else.

I was waiting for the three to actually have it integrated into releases that are out there. Then it would be good if folk filed a RedHat bug and got RedHat to integrate the later resolver into their releases. It would be good if all this happened before August to give us some data. But like anything in the DNS, there are a lot of variables going on there, and only Knot has this implemented at this point.

So I can't really push out a measurement ad at this point in time. There's just no results worth looking at.

WARREN KUMARI: A quick follow-up to what Geoff said. People who were watching [Sarah's] presentation from just before the break saw that there was a flat line of resolvers reporting RFC 8145 stuff. Then there's a huge spike where they all jump up to 30%. That big spike where they jump up is basically or seems to be people upgrading their versions of a resolver to deal with the security bug. This

means that what we need to do is wait for this to be deployed in resolvers and make sure that, shortly after that, somebody exposes a vulnerability so everybody will go through an update. So start looking now.

FREDERICO NEVES: Like not supporting [Sentinel] vulnerability.

JAAP AKKERHUIS: Jaap Akkerhuis from NLnet Labs. We've been waiting until the dust settles a little bit on Sentinel, and we're in the middle of the [unleash] for Unbound at this moment. As soon as that [inaudible] we were to put it in. So it's not a big deal. That's basically what it is.

WARREN KUMARI: Yeah. I think that we believe at this point that the core system and the label is finally stable. For folk who haven't been following the DNSOP list, the actual string that you choose – the kskroll-sentinel-is-ta (or no-ta) – has changed four or five times – yeah, five times, I guess. That was bikeshedding, not really an important change but something that had been changing. We think it's now stable, so implementers should be able to rely on that particular implementation or string.

RUSS MUNDY: Any more questions? Go ahead, Geoff.

GEOFF HUSTON: I want to just make one more comment about this. It illustrates why we're having a second bite of this and what was actually wrong with 8145 in the first place.

When you look at the DNS, you can look at it from two perspectives. One is actually trying to understand the behavior of individual elements in the DNS resolution system – in other words, individual resolvers – and trying to understand how individual resolvers are behaving.

But users don't do that. They have a set of resolvers listed in [etceteraresolve].com. If they don't like the first answer, they go to the second and go to the third. There is a real question about what happens to users as distinct from what resolvers are doing.

As an example, if you have two resolvers locally configured – one validates and one does not – the key roll is irrelevant because, from the validating resolver, even if they're not following a key roll, all they're going return back is SERVFAIL. SERVFAIL says, "Try the other." So the user is fine, even though some resolvers are not.

This Sentinel will not uncover recalcitrant resolvers. It's not intended to. It cannot do that. What it can say is, prior to a key roll, "What do we think is the population of users that might be left stranded with no working resolver?" So it's kind of the damage scenario. That's all it can measure.

So on Viktor's question – "What about resolvers? How can we fix this?" – if you want that kind of insight into the DNS, you've got to think a lot harder and make even more changes to either the protocol or resolvers because we don't have that ability today.

Thanks.

WARREN KUMARI: Go ahead, Jaap. One more question.

JAAP AKKERHUIS: The question that I have is: so we have a label that's deserved for this test specifically?

GEOFF HUSTON: The label excites a behavior. The domain in which that label is located is up to you and me and anyone else who wants to run this test. The label triggers a behavior. That's all.

WARREN KUMARI: Leftmost label.

GEOFF HUSTON: Leftmost label.

JAAP AKKERHUIS: So the question I had is, are there any other labels that exercise different behaviors? Is there an IANA registry somewhere of behavior labels? Because we're starting something. You're right.

UNIDENTIFIED MALE: [inaudible]

WARREN KUMARI: Yeah. There is xn--. Some people would say all of the underscore labels – stuff like _xmpp blah, blah, blah. They don't really excise different behavior in the DNS, but they excise some sort of different behavior. Also, certain resolvers like BIND and, actually, some stub resolvers won't allow you to use underscore labels for things, like if you use underscore [foo] with an A-record. Androids simply won't load it.

But, yeah, I understand where you're coming from. We've randomly chosen a string and started using it. This should probably document in some sort of thing somewhere. The obvious place would be: let's put it in the special use names

registry. But I think the registry is like, "Whoa, I don't want to touch that. Let's choose a different registry. [PANE], etc."

[FREDERICO NEVES]: Just one more comment. I forgot to say at the beginning that the session is translated in French and in Spanish. It's available through the headsets.

RUSS MUNDY: Well, let's thank Warren and Geoff for this information and the work and – oh, we have another question. Quick.

UNIDENTIFIED MALE: Just a comment on xn--. It doesn't affect DNS. It affects the next layer. So that's different.

RUSS MUNDY: Okay. Thank you very much, Warren and Geoff. Appreciate it.

The next person up is Ondrej. The clicker – here, we can bring you the clicker. Or you can do it there if you'd like.

UNIDENTIFIED FEMALE: [inaudible]. Yeah.

RUSS MUNDY: Would you rather use the clicker?

ONDREJ FILIP: Yeah.

RUSS MUNDY: Ondrej Filip from CZ.NIC will be our next presenter. We're going to hear about some of their experiences from their Turris Project.

Thank you, Ondrej. You're up.

ONDREJ FILIP: Good morning. Happy [P] Day. As said, my name is Ondrej Filip. I'm working for CZ.NIC. I would like to talk about one aspect of the project with is called the Turris Project because it's not just a normal CPE device. It has many other features. I'll give some time to introduce it a little bit.

What is the Turris Project? It started quite a while ago, in 2013. The main focus at that time was to create what had a very fancy name: The Project of Shared Cyber Defense. Basically, the idea was to create some CPE devices that would be spread out in the network, mainly at the edges of the network. They would collect some information to the center. We would somehow evaluate that information – we are also running additional [surteam] –

and then provide some [inaudible] rules and security recommendation and stuff like that.

So that was the basic idea. Then we continued work on this and that. We decided that the best device like that would be like the home SOHO router. So we started to work on that. Also, we added one more goal because the situation in this field is really bad. Many of those routers how very – how to say it nicely? – strange firmware. They usually do not have updates of the firmware, so there is some security bug. There's usually only a small possibility of fixing this. Also, the support of technologies like IPv6 and also DNSSEC validation was very bad.

So we continued on that. We hoped to find some device on the market to which we would just provide some addition to the firmware. Unfortunately, we couldn't find anything.

The result was that we created our own hardware. We created our own small router. For the purpose of the project, we created the first generation of the router. It's called Turris 1.0. We did exactly 1,000 of those devices.

Now you can see it on the screen. It's the blue router on the top left.

Later on, because the results were very interesting, we wanted to extend a little bit the project, so we created a new version

called Turris 1.1. Again we did 1,000 of them. The main focus at that time was the Czech Republic. We were giving those routers to people in the Czech Republic for free. A few of them were also given to people abroad, but the super majority of the routers were in the Czech Republic.

Then we started to publish the results of the project. Some people started to ask us, "Can we buy it?" "Can we get one?" and so on. Some were really trying to bribe me with credit cards and everything. Some were successful.

We decided to make this a little bit of a broader project. We went for a crowdfunding campaign because we weren't sure if this device could be sold on the market. It was a very successful campaign. We collected a lot of money and created this new router called Turris Omnia. This is the current router which we are kind of selling. But please keep in mind that we are a not-for-profit organization, so many of our activities are not really for [consumer] profit. This is our addition for how we wanted to make the situation in the SOHO field better.

What is different with Turris Omnia is that it is open-source, as everything we do. So it's not just open-source software, but also open-source hardware. The good point is that the hardware is very powerful. We have much more memory than the same

machines in this category. That's why we can do much more things on this hardware.

The operating system is called Turris OS, but basically it's a little bit [tuned to] OpenWRT. The main feature, which is a key thing for, is that it has automated updates. Whenever there is a problem, there's some security bug, or if you need to provide some, for example, new root key, we just issue an update. We do it quite often, actually. We don't just do security fixes, of course. We add new features. As I said, the hardware is quite powerful, so this device can be useful for many, many other things, not just the router.

As it started as a security project, security is the key point there and the main emphasis. This device shouldn't allow you to make some unsecure setup. It guides you through the configuration process. It doesn't allow you to have weak passwords or have some open ports and stuff like that.

Also, the communication with the center – the way how it gets the updates – is very well-designed. It has its own crypto chip with its own set of keys, so everything is hardware crypto.

It has many other features which are not related to this talk. It can run, for example, [as a] honeypot. Again, that's the way we collect information. It does flow analysis. It can find out, for

example, if in your [lander] there is some vulnerable device which is getting Internet or doing something unexpected.

As I said, we provide the results of the collected analysis, so it has an adaptive firewall, which changes according to the security situation. It can be run as a VPN server. And many, many other features and important stuff. It's fully IPv6-ready, and it's really very flexible. You can even run LXC containers, so you can run another server as a virtual machine inside this router, which shows how powerful it is.

The most important for us is fully doing DNSSEC validation by default. That's one of the sources of joy and pain in this project because, as I will explain later, this is not really easy.

One more remark. DNS is using quite a lot in this device. For example, we signal through the DNS which set of keys the device should use because, if there would be some security incident and some of the key set would be compromised, we can switch to another set of keys. DNS is important, so that's why even validation is important in this device.

When we started this, there was a lot of lessons learned. As I said, the operating system is developed constantly, so the situation at the beginning was very different. We were quite naïve. We just thought that rolling the device that will do DNS

validation was an easy thing, but the we found out that it's not so easy.

First of all, there are a lot of problems. I should point to some group of organizations that cause them. They are mainly ISPs. They are very creative in how they work with DNS and DNSSEC. So we very quickly learned that we needed to provide some additional GUIs and some page that informs the user what's happening and why the router is not working properly. So that was one of the first updates in the operating system.

The main problems are that ISPs usually have some broken DNS recursors. They set it up ten years ago and they believe that this is enough forever. Those very old implementations of DNS cause problems. Whenever, for example, you [set] as upstream recursor, that's a bad thing.

Even worse, sometimes they are very creative and install some middleboxes that do something magic with DNS traffic, mostly some bad magic. They had some reasons, probably, for this in their history. They probably wanted to fix some problems, they installed this box, fixed that particular problem, and forgot it, unfortunately. Some of them just redirect the whole Port 53 traffic to some recursor. So it's quite complicated to operate on this network if that resolver is broken. That's unfortunately often the case.

One thing which also is not just on the ISP side – it was a little bit of a problem when we started with the Knot resolver – actually, those blue books are the first one to use Unbound, but only as they use the Knot resolver. Such implementation in the wild was also a very, very interesting experience. We had some bugs in the resolver. But again, do to fact that we can instantly update the system, we were able to find a lot of bugs and fix them immediately. So now I think, mainly because of this project, the resolver is very well-tested software.

Last but not least, there are sometimes problems with some broken authoritative servers, mainly the EDNS issue. I think I will comment a little bit later on that.

This is the page that [we did a] little bit just for the DNS. As you can see, it's a very long description of what's happening. Of course, not many people use it. They'd rather call us anyway. But at least we try to educate them about everything that can be wrong with DNS. They have several options. They can either use or not use the upstream resolver. Also, unfortunately, we had to allow them to disable DNSSEC validation. If you click to not validate, you will get several warnings that you shouldn't do it or that you should do it temporarily. But unfortunately sometimes this is the last option for connectivity, so the people would rather switch it off rather than not be able to connect to the Internet.

We did several tests that showed what's wrong. Some people are brave enough and we're able to report these to the ISPs, so at least we fixed some of the issues at some ISPs' network. But some ISPs are very resistant, and they have a huge belief that they're perfect and that a DNS hack is the best solution in the world and should stay they forever. Unfortunately, sometimes it's a little bit [faithless]. So those two [tick] buttons solve a lot of issues and help a lot.

Let me conclude. Mostly the problems are at ISPs. They usually had some issues, either a problem with DNS or a security issues. Some introduce some solution and they forget that [said] solution. You need to discuss with them. That's not for the end user to be able to do this. Usually we try to help them, but of course, that doesn't work very often.

We had to introduce your configuration interface with tests. That helps a lot because at least power users understand the problem. If you send such a screenshot to your ISP, if they have an operator with a brain, they usually are able to understand what's happening. So that was a key point that decreased the number of calls to support.

The last thing, which was announced at DNS-OARC last week, is that, as you know, the developers of the open-source resolvers decided to sunset the workarounds in resolvers during 2019. At

least this is something we will start to ignore starting February 1st, 2019. We will not be the only ones. So, please, if you believe that you have some issues, try that webpage. Try it to test your domain and test if that domain name will be resolvable after February 2019.

If you allow me one more thing, we are just before launching a new version, by the way. This is going to be a new Turris. As you can see, it will be very, very flexible with many parts. You will be able to play with that module. Again, we are a not-for-profit organization, not really a hardware manufacturer, so we will again for Indiegogo campaign. If you like the Turris Project, do not forget to check Indiegogo during April and you will see this beautiful toy.

And that's all. Thank you very much.

RUSS MUNDY: Thank you, Ondrej. Jacques, go ahead. First question.

JACQUES LATOUR: By "flexible," do you mean we can grab it and just go like this?

ONDREJ FILIP: [inaudible]. It's modular. It's probably the better word. You could be able to design your own dream routers.

JACQUES LATOUR: Thanks.

VIKTOR DUKHOVNI: Are you maintaining any statistics on improvements in the ISP landscape? Any sort of graphs that you can publish over time?

ONDREJ FILIP: Yes. Of course, since we are a company based in the Czech Republic, our power is much, much bigger in the Czech Republic. Locally we were able to fix most of it. A phone call from the Czech Republic to a Belgium ISP doesn't help much. They just ignore it just because for them it's an exotic device. They do not fix the problem.

We estimate. It's really tough to say that roughly maybe a little less than one percent of the users have such problems, so they need to do something with DNS. That shows that the majority of ISPs are okay, or at least that the big ISPs are okay. But roughly one percent of users have some issue with that.

RUSS MUNDY: Go ahead, Frederico.

FREDERICO NEVES: Ondrej, are the old hardware versions compatible with the new software?

ONDREJ FILIP: Yes. They have the same operating system. Although they have a different CPU platforms, we still keep and maintain the system. Many of them are older than, I think, four years, and they still have absolutely new version of everything.

FREDERICO NEVES: Just a follow-up question. Do you guys any other platforms? Because it's based on OpenWRT. Any other hardware that runs OpenWRT can run the Omnia software?

ONDREJ FILIP: Not really. There is not many CPE devices with two gigabytes of memory on the market, to be honest. For many things we do, including the configuration system, automated updates, honeypots, and everything we need a little bit more memory than is normally available. But that's one thing we could probably discuss.

RUSS MUNDY: There's a mic back there. Or up here – yeah.

ABDALMONEM GALILA: Abdalmnem Galila, ICANN coach. I think that most people DNS administrators at ISPs don't prefer to do validation because they think that it is difficult to administrate DNSSEC or administrate the DNS servers that contain DNS validation. So when they use this Turris, it will decrease the administration side there?

ONDREJ FILIP: Yeah. They will not need to care anymore. But it's unrealistic to believe that 100% of ISP customers will have a single router or the same version of routers. So I don't think that's realistic to expect. But, yes, with Turris, the validation is closer to the end user. I think it will be ideal if the validation would be even closer on personal computers, mobile phones, and stuff like that. But with Turris, it's a little bit closer to the end users, so it's a little bit better.

RUSS MUNDY: Okay, Ondrej, I thank you – oh, was there another one I missed? Yes, please. There's a mic there and at the table.

UNIDENTIFIED MALE: [inaudible]

RUSS MUNDY: Get nice and close.

UNIDENTIFIED MALE: [inaudible]. I have a question. The title says DNSSEC validation is done at CPE. Is that okay?

ONDREJ FILIP: Why shouldn't it be okay? These are Consumer Premise Equipment devices – your home router – and the validation is performed there. So I believe that's okay.

UNIDENTIFIED MALE: [inaudible]

ONDREJ FILIP: The title is Validation at CPE. It's just Consumer Premise Equipment, which is the router in your home. So believe that fits the title.

UNIDENTIFIED MALE: Who validates?

ONDREJ FILIP: The router in your home.

UNIDENTIFIED MALE: Okay. Which software do you use in CPE?

ONDREJ FILIP: Do you mean or the DNS validation?

UNIDENTIFIED MALE: Yes.

ONDREJ FILIP: As I said, the older version used Unbound. All the Omnias – the vast majority of those devices – use the Knot resolver, the software we developed at CZ.NIC.

UNIDENTIFIED MALE: Okay. CPEs use Unbound or the Knot resolver? True?

ONDREJ FILIP: Yeah.

UNIDENTIFIED MALE: Okay. Thank you.

ONDREJ FILIP: Thank you.

RUSS MUNDY: I realized during this presentation that someone from my ISP at home is sitting at the front table who DNSSEC – Joe with Comcast. I use the Turris router at my home. So I have direct access if I ever have any problems. But I have to say that, in about three years of using the Turris at home and the DNSSEC validation by Comcast, I have not had one single problem. So it works very, very well. I just wanted to point that out to folks.

Thank you, Ondrej, or the presentation. I think we need to move on – real quick.

UNIDENTIFIED MALE: I'd like to combine the two topics for today, the KSK rollover and Turris. Could the rollover of the keys be done through Turris automatically, or should there be an intervention from the administration?

ONDREJ FILIP: No, because it's automated.

RUSS MUNDY: Thank you, Ondrej. Our next presenter for the morning session is Jake Zack from .ca/CIRA. Is he – oh. It is you. You're doing it. Oh, okay. Good for you. Thank you. It's Jacques Latour who's doing it.

JACQUES LATOUR: The clicker?

RUSS MUNDY: The clicker? Yes.

JACQUES LATOUR: All right. Jake is sick. He's got a nasty cold, so he left. I'll being doing this presentation on his behalf. I have speaking notes and everything. Here we go.

The talk is about the next generation signer that CIRA implemented recently. So we'll talk about that today.

A quick overview, similar to what we had this morning. We have 2.7 million domains. 2,400 are signed. 80% of the registrants are in this room, I believe. So not a lot happening there.

A couple of ICANN meetings ago, years ago, we presented our solution with a high-availability signer, BIND, and AEP Keyper. We got rid of all that and put in a new infrastructure in using OpenDNSSEC and the Gemalto HSM. So we did a KSK rollover and migrated our entire infrastructure with the new signing solution. That's the talk for today.

The setup that we built five years ago was a very high-availability infrastructure because we were mitigating the risk of

having DNSSEC failure. Back then, we generated the zone twice. We'd sign it with different signers, OpenDNSSEC and BIND. Then we did extensive zone validation. We compared one with the other and made sure that there were no differences between the zones. Then we published the zone live on the Internet.

When we put this in production the first year, we found a few bugs that could have been operational. The value of that system worked back when we didn't really trust the signer. The lessons learned is we learned a lot from that architecture and then we came up with a new system that' is way more efficient.

In terms of high availability back then, we used Oracle at hot standby, so the backup site was always cold with the up-to-date data. So we had to turn it on. The process to go to the backup site was a little bit complicated.

We had an active/passive Oracle database for the registry. We used OpenDNSSEC, the beta version, when it was installed. That was the same version that we use in production. We never upgraded that version. If it doesn't' break, don't mess with it. So we didn't.

We had a lot of issues internally with validation, the zone, the empty node terminal issue, and all that. Over the years, we've built customization to make sure that our legacy setup worked.

The process took about 35-36 minutes to generate the zone. It was based on CRON. We generate the zone, sign it, do a bunch of validation, a lot of file copying, SEP of the zone file to compare the one with BIND and the one with OpenDNSSEC, and make sure it's all that. We copied the zone at our backup site and signed them also at our backup site in real-time and compare the whole thing. So it was a very extensive data manipulation process. We've optimized that with time.

With this system, we never had a zone outage from our signer solution over time with all the redundancy. Back then, we didn't have any automatic failover from site to site. That's something we wanted. Everything was very [inaudible], so all the signers, all the validators were blades/servers with OS and a lot of copying and lot of processing to do our zone. So that was way too much.

The other thing is that we were using AEP Keyper. They were running on a five-year limit. Everybody was saying they're going to start to die, so we looked at replacing that with much better load balancers and a much better process. For the AEP Keyper, you need to manually go on-site and put in cards. They were good at the time, but today there are way more modern HSMs that can be more efficient and offer the same security.

On the validation part, in the beginning, like I said, in comparing both zones, we found a lot of bugs and issues and problematic

implementation with .ca. Within our .ca, we also signed third-level domains because we have provincial. So we have on.ca, qc.ca – all provinces. Then we have delegation at the third level. That caused issues with zone validation. We also legacy for four levels of zones. We have city and province when you add the DS at the four levels. But the city is not signed, so not a lot of validators support that type of infrastructure. It took a while to fix those issues.

The other thing is that BIND and ODS sign differently. Jake had to manually once in a while go in and clear the signer because we're comparing BIND and OpenDNSSEC. Once in a while, he had to do manual intervention to restart our signer, especially during a Super Bowl weekend. Jake didn't like that.

The new setup is much, much simpler. We went from 16 physical machines to 8 VMs. We could have went to 4 VMs to make it simpler, but we decided to have the zone distributor on their own instance. All of these boxes are in different security zones, all firewalled internally. So it's much more of a streamlined architecture. We have zone generation, the signer, and the validator all in one box. Then we have two for availabilities and two zone distributions. It just works.

We did a bunch of evaluation and we decided to use – I'm not sure what they are. It's [Tail], the [bot] Gemalto, the [bot] Luna,

and the [bot] Safenet. So it's one of those HSM [picker] companies. People call them different names. We call them Gemalto. We like them because you can do pretty much anything you want with them. You have partitions.

The main reason we went from AEP Keyper to Gemalto is that the signer solution we have isn't integrated in a new service we want to build, which is a DNSSEC signing service. We want to be able to use this infrastructure to suck in zones from .ca registrants eventually; so, second-level zone and sign in here and return the signed zone to the customer. And we don't want to have to replicate the entire infrastructure. Gemalto is one of good infrastructures to support this.

There's a lot of capacity. You can have 20 partitions and up to five years of keys pre-generated in each of the partitions. So they're very flexible to support.

The other thing is that it was much easier with Gemalto to do the key signing ceremony. With AEP Keyper, it took forever. With Gemalto and the process they have, it's an hour instead of a few hours of the key ceremony. So it's easier on the process.

The other thing we did was we switched from the Oracle standard to the Oracle ODA and Dataguard. That's the Oracle appliance. It's actually/active, so that's pretty neat for the registry. That means we generate the DNSSEC signer. It

generates active/active on both sites. Then we distribute our zone pretty much in real-time from both the backup and the primary site.

The note here is: our backup site is less than one second behind the protection site. So this works great. It's a little bit expensive, but it works.

We don't dual-sign the zone anymore. We pick ODS. That was the decision. So we don't have any issues with empty node terminals anymore. We use Springbatch and orchestration and all that. The zone generation works much better now with the signing.

The HSMs are low balance. Before, we had an HSM dedicated to a server. Now they're just a pool of HSMs available for signing.

We use the same HSM security control framework and then we build for the initial key-signing ceremony. Here's a word you haven't heard in a long time: DPS; define the structure for managing the keys. We have different – we have special slides here – people in the office with different roles that can do different things on the HSM, and Gemalto to support that in the same way that the AEP Keyper does. We have crypto officers. We have witnesses. We have admin. Certain people have access to rooms and other [nodes].

With AEP Keyper, we were able to replicate and maintain the same DPS (our DNSSEC Practice Statement) that we had in the past. All of that works with Gemalto. So that was a good thing. We didn't have to reinvent our key signing ceremony process.

It's much easier with the thumb drive to do the ceremony. They have a keypad and thumb drive that you can use to manage crypto-ceremony. That was more efficient.

At the same time that we did the HSM key rollover, we had to do a KSK rollover because we had a new KSK in the HSM. We had the old KSK in AEP Keyper in the old one. The new key was in the new HSM. We took it slow to do the KSK rollover and to do the HSM rollover.

`

At the same time, we migrated the entire process. It's pretty hard to see here, but we have the KSK that was in production. We introduced a new KSK in the zone with the new signer. Over time, we migrated. So we had one key. We had both keys. We had the new key signed with the old key, and then we did the switchover of the HSM, and then the old key was signed with the new key, and so on. All the IANA changes were done. So it was pretty much straightforward key signing/key rollover process.

At one point in time, we talked about doing a protocol change. Based on what we learned at DNS-OARC, that would have been a

bad thing, I believe. So that was enough, just to do the KSK and the migration.

So we took it slow. We waited weeks between every step. There was no rush to do this. We have multiple KSKs and multiple ZSKs, so it got bigger at one point in time. But the entire process actually worked well. We didn't have any outages.

So we got over. It went smoothly. Nobody noticed or e-mailed us or had any signing issues. The cool thing is that we used to generate the zone file before on an hourly basis. Now we're down to 30 minutes. It actually takes 11 minutes to do it, so we could even go faster. But we want to wait a little bit to do that.

On the KSK process with the new KSK ceremony, we can remotely manage the HSM. It's minutes of process instead of manually having to go on-site and play with smart cards and HSMs. So that history is gone.

Perl – poor Perl – is gone. No more.

So that's it for the process. There are no notes here, so…

Questions?

RUSS MUNDY: Thanks, Jacques. Do we have questions?

Go ahead, Robert.

ROBERT MARTIN-LEGENE: Hi. This is Robert Martin-Legene from PCH. I think it was [inaudible], if you want to speak in Spanish. Never mind.

You mentioned something about that the old HSMs and the new HSMs were very different in terms of management. It sounded like you got a lot of benefits just from switching and you didn't need extra hardware for backup and stuff. Is that just because you switched, or is that just because an optimization you did at the same time? Because you had that thing – oh. Two-fold.

That one. You have a Gemalto backup unit at the same security level, and before you needed an offline full HSM. You don't need an HSM now?

JACQUES LATOUR: Before, we had the offline HSM. We did a key signing ceremony in our office with the offline HSM that was in a safe. We generated the new keys on that, and then used the smart card out of that. Then you grab the smart card and you go into all the other HSMs and reprogram them.

With Gemalto, they have a framework where you can have a control, some sort of a keypad, that you can use to remotely manage all the HSMs. You can push keys and change keys securely from a central location. You need to set up, I think, a

black key that creates trust amongst all the HSMs. From our office, we can actually reconfigure all the HSMs using that keypad. So the offline HSM was in a safe that only your lawyer could access. Now we have the keypad in a safe that's protected.

ROBERT MARTIN-LEGENE: Okay. [PCH uses] the HSM [inaudible] said you used to have. We don't travel around the world to load keys. I don't know why you need to do that.

JACQUES LATOUR: While you're offline, all your HSMs are offline, right?

ROBERT MARTIN-LEGENE: All our KSK HSMs are offline, yes.

JACQUES LATOUR: Our KSKs are live here. If we want to do anything – add a new customer or do whatever – we can do it right there and then. We don't need to wait for a key-signing ceremony like you do on a quarterly basis to add and delete customers. No pun intended.

ROBERT MARTIN-LEGENE: Right. There are pros and cons against on both sides, right?

Okay. Do you have a video of your key ceremonies or something? Is that something you publish?

JACQUES LATOUR: It's not published, but if you really want to see it…

ROBERT MARTIN-LEGENE: I really want to see it.

JACQUES LATOUR: It's going to cost a beer.

ROBERT-MARTIN-LEGENE: Go ahead, Joe.

JOE ABLEY: I was just going to make a comment. I remember the early days of when you first deployed DNSSEC. A lot of the processes were modeled on what happened in the root zone, including the model of HSMs.

I think it was true at one point that that was the only kind of HSM that was certified to the level that the U.S. Department of Commerce required ICANN to use. So there wasn't a lot of choice at that time. It's good that things have moved on.

But it sounds like what you've done is evolve the process from what was appropriate to the root zone with its heavy focus on physical security to something that's more appropriate to operations. That I think is a good message. Because lots of people copied ICANN when designing their procedures around this stuff. So I think that evolution is good to hear about.

UNIDENTIFIED MALE: Viktor – yeah.

UNIDENTIFIED MALE: [inaudible] from AFNIC. I have a small question about the process. My first question is more for clarification. You said that, with the new process, it takes about 11 minutes. Do you still a complete zone file generation?

JACQUES LATOUR: Yes.

UNIDENTIFIED MALE: And did you consider to use something more dynamic?

JACQUES LATOUR: Yes.

UNIDENTIFIED MALE: Why didn't you do that?

JACQUES LATOUR: Well, for .ca it takes 11 minutes to do the full zone. If we're able to keep the process we have within 15 minutes, it's good. But we only have 2,000 signed delegations. Once we're in the millions, then we'll revise –

UNIDENTIFIED MALE: That was my second question. If you are to do the same process with, for instance, 10 persons or 20 persons on the signed delegation, does it scale?

JACQUES LATOUR: Eventually, if we get adoption in .ca for DNSSEC, then we'll have a dynamic solution.

UNIDENTIFIED MALE: The last question – sorry, I'll try to be as fast as possible – did you some tests with, for instance, larger ZSKs? Because I know that the signing process is longer with 2048, for instance. Did you test that, and does it scale?

JACQUES LATOUR: I don't think that we did, but the performance level between Gemalto and the AEP is more than ten times faster. So that's [inaudible]. We haven't done –

UNIDENTIFIED MALE: Okay. I said that because I know there is a factor between Gemalto and AEP. With the different size of keys, sometimes the difference is not much bigger. So you should do some tests on that.

JACQUES LATOUR: We didn't play with that yet.

UNIDENTIFIED MALE: Thank you.

RUSS MUNDY: Thank you very much, Jacques. Our next presenter is Joe Crowe from Comcast, who's going to tell us about negative trust anchors.

Thanks. You're up, Joe.

UNIDENTIFIED MALE: Russ, a pretty small comment, actually. We are comparing different stuff because, as far as I know, the Keyper is still the

only level for validated HSM in the market. The Gemalto is a level three.

JACQUES LATOUR: And we have no requirement for certification.

UNIDENTIFIED MALE: No, no. That's not the comment. I'm just commenting on Joe's comment regarding…

JACQUES LATOUR: Yes.

JOE CROWE: All right. Afternoon. Joe Crowe from Comcast, Senior Engineer. We do DNS, NTP, and DHCP. We're a core network services team.

At Comcast, we do about 500 billion queries a day. We've been doing DNSSEC validation since 2012. DNSSEC does scale, as I mentioned earlier. If you're out there, enable DNSSEC validation. It's pretty easy.

What does that mean for Comcast? Well, when DNSSEC validation fails, we get the blame. Customers believe we're blocking websites. When a big company like us does DNSSEC validation, customers quickly reach out when they can't reach

their favorite site. I know NASA.gov is usually the one instance everybody usually talks about.

The one here is HBO Now. As soon as they went live with HBO Now, their DNSSEC validation broke. Their DNSSEC was broken, we couldn't validate correctly, and Twitter blew up. Net neutrality, we're breaking you. We're blocking you. Most of the fixes are that people would switch to a non-validating resolver. Usually they tried to switch to Google. But guess what? Google would be doing the same thing since they are doing DNSSEC validation as well.

One of the temporary fixes that we can do is put in a negative trust anchor. Why would you use a negative trust anchor? Well, if you're too big to fail, then there's that instance of security issue. "Okay. Do we know if this domain – NASA.gov, state.gov, sorry, it's a lot of .govs sometimes – is actually a security issue, or is actually an operational issue?"

Well, most of the time, our experiences have shown that it is an operational issue. Very rarely do we actually notice a security issue. Actually, in the time that I've been at Comcast, I don't think I've really ever seen a security issue. It's all been operational issues.

What do we have to do at that point? A few options: just let the domain continue to fail, turn off DNSSEC validation – hell no,

we're not doing that – or just put in a negative trust anchor for that one certain domain and allow that to continue to resolve on a resolver, with the known fact that it is failing DNSSEC validation in the backend.

When you first want to start doing negative trust anchors, you want to come up with a good process. You want to get together with your team and find out when and why you want to do an NTA. You want to also know how you can implement on all of your resolvers. Internally, we use multiple vendors. We want to make sure we have a process and know the process manually for all the vendors because, if your automation is failing for any reason, you still want to be able to implement this. And then stay consistent with that process. If there's something that had changed with how your vendor is implementing NTAs, make sure that you know that and you have updated everything correctly within your automation and your team knows what you're supposed to be doing.

We want to know what the risks are of keeping that domain to fail. Is there going to be cost associated with that? Is it going to be a long time that that domain is failed? The more calls that we get in Comcast, the more it would cost us. So is it really failing? We want to know. Is it failing because of an operational issue?

At that time, we'll do our checks or troubleshooting and make sure that we reach out to the right people. We'll either do it by e-mail or by Twitter, letting people know that we have seen a domain fail and that we're going to let it fail and we're reached out to the correct people. We do have a @Comcast Twitter handle, which is actually used and monitored by our team.

Then automate. You want to make sure that, if you're touching more than 25 servers – more than two in reality – you're automating everything. The negative trust anchor tends to have to be approved in our org by senior leadership. If we can prove to our senior leadership that, yes, it is failing because of an operational issue and we've reached out to the correct people but it has taken more than 15-20 minutes or 30 minutes to actually get a response back, if it's a big site like NASA.gov, state.gov, or HBO Now, things like that, we would actually implement an NTA because the costs associated with that would be better than actually letting it fail. But we've gone through the correct steps. Again, that would be a senior leadership call at that point.

Automation helps us scale. In our case, we run multiple vendors with different commands to implement our NTAs. If there's an error, there's an error in one spot and it's broken across the board. It's broken the same way, we know it's broken the same way, and we can fix it pretty easily at that point. Just like any

good operations team, you should actually be doing your testing in a lab. Everybody says, "I do all my testing in production." That's not the best place to do it.

In our basic automation at Comcast, we utilize SaltStack for our automation tools. With that, we have pillar data that allows us to update one spot with the domain that we know is failing. That allows the pillar data to be consumed by more than one vendor and one script. We have one command that's run from a Salt Master that can push to our hundreds and hundreds of servers. We would usually do it in a rolling push. That way, we can test in one area and make sure everything pushed correctly and then just roll out completely everywhere. It'll usually take anywhere between a couple minutes to ten minutes, depending on how fast we really want to get it out, if we really want to get it out, if we want to roll it and test it, or if we've tested it enough and we know that we that it'll push correctly with no errors.

Our basic structure for our pillar data, as you can see on the right here, is the YAML format. We have an internal NTA and a customer NTA. Our internal NTA is actually going to be a lot bigger. A lot of old bad practices internally can break DNSSEC. There's improper delegations and [dotted] host records in places there shouldn't be. When we're trying to implement DNSSEC internally, that's been one of our biggest hurdles.

When we apply the NTA state to all of our resolvers and, like I said, the above YAML format, that data will be consumed by scripts, consumed by another file. Then we do our testing after that.

Most everybody here should know basic troubleshooting. DNSSEC failure will result in a SERVFAIL, appending a +cd. Can you dig it? You want to make sure you can dig. That's the first step. Testing with other DNSSEC-validating resolvers ensures that it's not just your resolvers. DNSviz is your friend. We actually own DNSSEC-failed.org, which purposely has broken DNSSEC. If you wanted to turn on DNSSEC validation on your resolvers, you can actually use that as a testing domain to ensure that DNSSEC is validating, or, in this case, not validating correctly.

In case for a key rollover issue for some sites, we would just do a cache flush across our board, and that usually will fix DNSSEC failures at that time.

One thing that came out of the conversation from OARC is how we could share NTAs from a big perspective like Google's, Comcast's, and other DNSSEC validators'. One thing that internally we've talked about is actually automating your own zones and using DNSviz's CLI. You can actually load in your own zones and do checks with a Cron job and say, "Are my zones working correctly with DNSSEC?" If they fail, it e-mails you or

alerts you any way that you want. At that time, you can find that issue before it actually gets out to the wild. TTLs sometimes are your friends.

In this case, yesterday's state.gov was failing for a little bit. After a while, they fixed their stuff, and we were still holding onto old data. It was a simple cache flush for us to actually fix it.

That's it for my talk today. If there's any questions…

RUSS MUNDY: Questions for Joe?

Yes, Paul? Find a mic and you can say who you are.

PAUL WOUTERS: As [John Gilmore] wisely said, if you can't not trust your friends, who can you not trust? In this case, we wouldn't know if you placed a negative trust anchor in there for malicious purposes or because, let's say, the imaginary government puts a gun to your head and says, "You must do this because we demand it." Do you in some way publish negative trust anchors that you have inserted?

Of course, how do you then avoid having a wall of shame, which also nobody really wants? It would be nice if there was some way of auditing what you're doing.

JOE CROWE: Currently we do not have anything that published the NTAs we have put in. I agree with you, the wall of shame sometimes is not the best way to call people out and say, "Your stuff is broken." Unfortunately, at the scale of Comcast, we have to let our customers know that we're aware of an issue. That's where our @ComcastDNS Twitter handle comes in handy because we can pretty much say, "We are aware that this website is failing DNSSEC validation. We're giving them a chance to fix it and then go from there."

As far as trying to publish the NTA records, I was actually thinking about this the past couple of days. I think that would have to be at a community at-large type of thing, where we all get together and say, "This is how we'd like to publish something like that." I don't think that a company like Comcast should be the one that dictates that type of thing.

RUSS MUNDY: Other questions for Joe? Robert, go ahead.

ROBERT MARTIN-LEGENE: What is DNSSEC is enabled on Twitter.com and it fails?

JOE CROWE: You're SOL. Yeah, don't know. At that point, we would try to find another way. Our frontline support would know that DNSSEC is failing for bigger domains. Unfortunately we're not really worried about the smaller domains. We're worried about business partners and websites that can really just cause that uproar from people.

VIKTOR DUKHOVNI: I have one last question, I hope. I know that your mail team has a similar process for whitelisting DANE domains that are failing. Do these go through the same kind of logistics, or are they completely separate?

JOE CROWE: That's completely separate. The mail team has their own process of whitelisting. That's a different process than what we currently do.

UNIDENTIFIED FEMALE: We've got until…

RUSS MUNDY: 12:30, right?

UNIDENTIFIED FEMALE: No. 12:15.

RUSS MUNDY: 12:15.

UNIDENTIFIED MALE: 12:15.

RUSS MUNDY: We have some folks coming in for a meeting that starts at 12:15. Is there a scheduling faux paus here?

UNIDENTIFIED FEMALE: I think so.

RUSS MUNDY: Oh my. Well…

UNIDENTIFIED FEMALE: [inaudible]

RUSS MUNDY: Okay. Our lunch isn't going to ready until later. Meeting room conflict. I guess there's a scheduling error.

UNIDENTIFIED FEMALE: Yeah. It looks like – everywhere it shows them right now. [inaudible]

RUSS MUNDY: Okay. So we had a change that our Workshop Program Committee didn't know about. We will have to go ahead and clear the room. Our lunch is scheduled for about 12:15 or 12:20. We can wander up there slowly and do the quiz after we come back if there's still time.

UNIDENTIFIED MALE: One request I have –

RUSS MUNDY: The tickets are these. Be sure you have your ticket when you head up. The lunch is on the terrace on the third floor. Please, everyone, go ahead and pick up your hardware and your lunch tickets, and take your other papers if you don't mind so it'll be clean room for the folks that have it during lunch.

UNIDENTIFIED MALE: I have a request.

RUSS MUNDY: 1:30 is our start time.

UNIDENTIFIED MALE: Okay. I have a request that people be on time. I have a lot of slides, so I don't want to start late.

UNIDENTIFIED MALE: I also have a request. Can someone check that we do actually have the meeting room at 1:30?

RUSS MUNDY: Yes. Back here at 1:30.

UNIDENTIFIED FEMALE: But their session goes to 1:30.

[END OF TRANSCRIPTION]