ABU DHABI – DNSSEC Workshop - Part II
Wednesday, November 1, 2017 – 10:30 to 12:00 GST
ICANN60 | Abu Dhabi, United Arab Emirates

UNIDENTIFIED FEMALE: November 1st, 2017, Hall A, Section BC, DNSSEC Workshop Part 2, 10:30 – noon.

UNIDENTIFIED MALE: Okay. Should we start?

UNIDENTIFIED MALE: You're moderating this session?

UNIDENTIFIED MALE: Well, Russ is.

UNIDENTIFIED MALE: I just want to know how much time I have.

RUSS MUNDY: Okay. We're about to restart again with our next panel presentation. We're going to slightly change the order from what's on the program – is this yours, [Barry]? – and – whoops. I thought I saw – oh, Duane is over there. Yeah, why don't you come up here? A little more visible. And – yeah. Right.

Our revised order: Duane Wessels will go first, followed by Jaap Akkerhuis and Cristian Hesselman, Geoff Huston, and then Roy Arends with ICANN.

I think we have sufficient time here, but I may try to shove folks along if we end up getting stuck somewhere along the way so we can make it through and have sufficient time for Q&A on each of the presentations. So we will take some questions with each presentation. I think, with that, we'll pass it to Duane. Go right ahead, please.

DUANE WESSELS: Thanks, Russ. I asked Russ if I could go first because this is sort of the background for the later presenters, I think.

This presentation is about RFC 8145 trust anchor signaling data. My interest in this was both as co-author to that RFC and also as someone who receives some of the data. This is a shorter version of a talk I gave about a month ago at DNS-OARC. That's why it says the good parts here in the title.

This RFC defines the process by which validators can signal their trust anchor knowledge. I'll briefly explain how this works. The signal takes the form of a key tag, which is a 16-bit integer that identifies keys and signatures and things like that.

There are two key tag values that are relevant for the discussion today. 19036 is the key tag for what we call KSK-2010, and 20326 for KSK-2017. These signals are reported up to the zones' authoritative name servers, which in this case is the root zone. The validators should transmit them about once every TTL or so.

The RFC defines two forms of key tag signaling. One of them is an EDNS0 option. In practice, this didn't get implemented by anyone, so, really, all the data comes in the form of the second format, which is a key tag query. It's just a normal query where the key tag data is encoded in the query name and the key tag values are encoded in hexadecimal. It may helpful to know that 19036 is 4-alpha-5-charlie, and 20326 is 4-foxtrot-66 in hexadecimal.

This table shows a timeline of how things rolled out. The first Internet draft was December 2015. By 2016, there was a first implementation in BIND. Then there was a later Internet draft, which became RFC8145 in April. At that same time, it was implemented in Unbound. May is the time at which I started looking at the data.

In BIND, this feature was enabled by default. In Unbound, it was initially not enabled by default, but it did change to default in early October.

Some data. As I said, the signal data is sent to the root zone name servers. The data that I have to look at comes from two of those – from A-root and J-root. I want to be clear that this data comes from only recent implementations of name server software, so only people who have recently upgraded are providing this data.

This slide shows what the raw data looks like. You can see these columns here at the bottom. There's a UNIX time stamp, [its] query name, where you see underscore-TA-hyphen and then the hex digits, then source address, destination address, and year, month, and date.

This sample was taken from earlier on in the collection. Most of the columns here are showing only the old trust anchor, which is the 4-alpha-5-charlie. Most of these lines have only that. Some of them have both values, indicating that these are sources who have already gotten the new KSK.

This graph shows the number of IP addresses providing the data per day over the time period, starting in May and until just very recently. The first vertical line on the left marks the time at which KSK-2017 was first published in the root zone. The second vertical line on the right indicates the time at which the RFC5011 hold-down timer ends. According to that protocol, that's when validators can add the new key to their trust anchor store.

Initially, we had something like 500 sources per day. Very recently now we're up to 2,500 per day.

This graph shows which signals those sources are sending. Again, this is per day. The red here, which is the blob in the left, are signals indicating only the 2010 trust anchor. The large green on the right is sources that have both 2010 and 2017 keys in their trust anchor set.

It may be a little bit hard to see, but in between the red and the green, there's a smattering of yellow pixels here. These represent source IP addresses that, over the course of the day, sent mixed signals. At some times they said they have only the old key, and sometimes they said have both the old and the new key. But that's a relatively small percentage.

The thing that was really concerning for the rollover was that, after that hold-down timer ended, the red line stayed flat and didn't go down.

Okay. Thanks, [Jaap]. You'll notice on the very far left that red line. We see another uptick. I'll talk a little bit more about that here.

This is really the same data, but it's represented as a percentage instead of a count. You can see that, right after the hold-down timer ended, there was this big crossover in which a lot of the validators our there indicated that they accepted the new trust anchor and converted over.

I don't have a point, or I don't think, but if you look down again at that lower right corner, towards the end of October you this uptick. Preliminary indications are that this is due to Unbound software. Unbound released a version on October 10th which enabled the reporting of the trust anchor data by default. It seems that that population that is using Unbound – maybe a significant number of them didn't have the new key.

Thanks, [Jacques].

Another thing that I thought was interesting to look at is how often we see unexpected or non-IANA key tag data in this. Since the start of the collection, I see about 29 key tags other than the two that we would expect as IANA trust anchors. When I gave this presentation at DNS-OARC a month ago, that was 19. So it's gone up a little bit since then. Similarly, back then it was only from about 12 distinct source IP addresses per day, but now it's up to 100.

Here's a graph that shows the unexpected key tag signals per day. Again, down at the lower right there we see this uptick. Again, my guess at this point is that these are being provided from the Unbound software. But the reason that the blue line went up a lot is because it seems to be that some decent-size population – maybe in particular operating systems or something – did something I would consider silly. They added the root zone ZSK, the zone signing key, to the trust anchor set, which was harmless. But it results in this uptick here. It's a little bit puzzling why that happened, but that's the best guess.

My conclusion from this is that this data is of reasonably good quality. I didn't see any evidence of tampering or anything like that. NATs and name servers forwarding to another name server and dynamic IPs and also having only visibility into two of the root servers makes this analysis complicated. To get a better picture, it's better to have more root servers. I think Roy will be talking about that later.

I already mentioned about the [strangeness] with October 10th, which, again, I think is due to data coming from Unbound users.

I'd like to thank ISC for implementing this and turning it on by default, and also NLnet Labs for the same thing: for implementing it and turning it on in Unbound. I would encourage other software vendors to also provide this data going forward.

That's the end of the presentation. Are we doing questions now or later?

RUSS MUNDY: Yeah. We got time for a couple quick ones if there's any at this point.

DUANE WESSELS: Okay.

RUSS MUNDY: Yes?

DUANE WESSELS: Ondrej? Oh, sorry.

STEVEN BARRY: Steven Barry from .ca. You attributed a couple of the variances in signal uptick to implementations in Unbound. I may have missed it, but do you know the relative numbers of Unbound versus BIND implementations that are signaling?

DUANE WESSELS: I didn't have time to include this, but in the DNS-OARC, there's some slides that show how we know this. The best way is that BIND provides the signals at very regular, 24-hour intervals. Unbound – I ran both at home and looked at them – provided the data at less regular and shorter intervals. We can see that in this data, and that's one of the reasons that I attribute these recent signals to Unbound.

As to the percentage, I don't have an exact number. It's maybe in the 5-10% range or something like that.

Ondrej?

ONDREJ SURY: Thank you for the presentation. Just to show that your request was hard, we just implemented this feature yesterday. Basically it will go to the next release and also the KNOT resolver will support this functionality in the next release.

DUANE WESSELS: Excellent. Thank you very much.

ROY ARENDS: Quick question, Ondrej, when you schedule that next release?

ONDREJ SURY: Honestly, no. I'm sorry. It was just coded yesterday. It didn't go through the code review and testing and everything. So it will take some time. We cannot release something that is untested, unfortunately. But we will try to speed it up as much as possible. I'll push the developers to make it quickly.

RUSS MUNDY: Go ahead.

ROY ARENDS: Duane, one clarification. I know we talked about it yesterday, but you say key tags other than 4a5c and 4f66. But I think what we've seen is an addition of a key tag to 4a5c and 4f66. That means the signal that we saw does have the two keys and another one. Is that correct?

DUANE WESSELS: Yeah, that is correct. Again, that uptick at the right there – those do have the expected keys plus an unexpected key. So the blue line there indicates signalers that provide the expected keys and the unexpected keys. The red line, which is solidly down at the bottom, is only unexpected key tag values from those sources. So these are other experiments or something else.

RUSS MUNDY: Okay. I think we need to move onto our next presentation. Thank you very much, Duane.

Next up is Jaap Akkerhuis.

JAAP AKKERHUIS: Jaap Akkerhuis, NLnet Labs, and doing Unbound as well, among other things. I think talking [inaudible]. It's very interesting. It also shows that the signal you get is very difficult to interpret. You don't know what's happening. It might just as well be keys escaping [inaudible].

Anyway, before that, I will talk a little bit about Unbound and 5011 and how this happens if you're a vendor who is implementing stuff that you don't know what it is. So that's some background.

In general, new features pop up in DNS. The policy is that we are conservative in putting the code out. Basically what we do is, whenever an Internet [inaudible] started, we think the ID will actually survive and not die a horrible death, if there's enough time, we will actually have in the code base or in the internal code some versions of getting stuff done, probably commenting on the new feature, whether or not it's feasible and does anything like that. Once the specification gets a little more stable, it will be released in a [standard] release, but only as a compile [time] option. So people will really know what to do. People who know what they're doing actually might be able to experiment with this, but they have to do some extra work for [inaudible].

Then there is this 48-hour period in the IETF. That's when the draft is getting stable and only nitpicking comments are put in. Well, 48 hours might take a couple of months in IETF terms. So depending on how long the 48 hours takes, it will actually put in [to regarding] the release as a runtime option. So the WSIS default is switched off. So it becomes a runtime option. People really wanting to be living on the edge of the world can actually already switch it on if they know what they're doing.

Whenever the standards really came out, as in RFCs, we went to the 48-hour period. We basically take off whatever is recommended in the RFC and put it in as a default setting. We might sometimes, when there's a real big change, wait until not the next release but a major bump of the release [are coming from] seconds because it's actually a new feature to functionality. We try to keep that automatically. So that's the standard policy. Some others are doing it as well. [Took resolver from] [inaudible].

How did it happen with 5011? Well, 5011 actually is a separate program. It's called Unbound Anchor, which is completely different from what it is now. So it was separate so people could play with it. It was actually [not touching code] at all. But when it went into the 48-hour period, we moved it to the main code base. We had incorporated in Unbound 1.4.0 in 2009 already, so even before DNSSEC was implemented in the root. But it was there. This is [inaudible].

The problem is it's a weird protocol. Actually, 5011 is not a protocol. It's just documentation of state changes. It doesn't tell you what to do. It only tells you how to do stuff. That makes testing seem like that's fairly [hard]. The code base for years already has had what we call the uni test of [inaudible]. We can test separate functionalities, and it's always run before we release anything major.

For doing 5011, we actually created – at least [Volker] did most of the work – a test harness – something happened to its spelling there. Anyway, that's what you get when you do slides at the last moment. The test harness is a uni test which actually accelerates the time, so you can pretend that things happened more quickly. That's to come [up] with a 30-day time period, which is mentioned in 5011.

Actually, the whole idea of doing this test harness has been taken up by the guys in cz.nic, and they made it slightly more generic. They actually have released it as a separate blob of software.

After 2009, some incremental changes happened, like bug fixes, because, if other features are implemented, it might require code changes I how you deal with 5011. But nothing essential changed.

So it's been [lying there], and we actually thought about doing T-shirts supporting 5011 since 2009, but we never did.

Anyway, then because of the accelerated testbeds – this is the Rick-Roll which Rick Lamb set up and the key rollover by Warren Kumari, which actually do rollovers all the time. I forgot how many times a day Rick does it. Warren does it every 90 minutes, I believe, which is kind of quick, depending on 30 days.

To test it, we had to change the code in Unbound to actually follow this fake protocol. What happened there was that – this is actually not really nice because you'll certainly have an extra code [pass] in your code which will be never be used in any work. And the real code [pass] will never be tested. Every [family] has that.

Due to the fast timing, we actually missed a corner case. The corner case, now fixed again, is, whenever you start to do DNSSEC and 5011 in a period which is actually shorter than the 30 days required in the protocol, you don't know what to do. Being very strict – actually the Unbound [fallout] the protocol to the latter. So [inaudible] was only noticed whenever the 30 days were over.

BIND actually basically blindly says, “Okay. Fine. This is a rollover. Let’s escape the 30 days.” We have a slightly more fancy way of checking this stuff, but in that case, we noticed that this is happening. This is the [inaudible] non-specified [inaudible] protocol, so we do that. Anybody with an Unbound which is older than that – there’s quite a lot of them around because a lot of people like [inaudible] – are not really following the latest code pass, but they’re backporting security stuff in all the versions. So they might or might not do 5011, which is another problem, with trying to find it.

If you are doing DNSSEC in the time before 30 days, you’re fine. If you are in this unlucky situation, the only thing you need to do is just remove the old anchor and restart Unbound, and things will be fine as well. So it’s not a big deal. Anyway, the latest version of Unbound now recognizes this. That’s what I’m saying here. If you really have [inaudible] throw away the old KSK and restart the server. You’ll be happy.

What it actually shows you here is that nobody really looks at the [inaudible] of this new protocol of what’s going to happen. You only find that in reality.

Anyway, the Internet is just a big experiment, so we really can’t replace it with something else.

RUSS MUNDY: Okay. Thanks, Jaap. We'll have time for a couple of quick questions for Jaap if there are any. We have maybe a couple minutes if there's a couple quick questions.

I don't see any. Is there anything online?

Okay. In that case, thank you very much, Jaap. We appreciate it.

Next up is Cristian Hesselman on the Root Canary project.

CRISTIAN HESSELMAN: 23 minutes. I won't need that much time. All right.

I'm waiting for the slides to pull up.

I'm going to briefly talk about the Root Canary project, which is a joint effort of several parties, as you can see on this slide. I'm with SIDN, which is the folks on the lefthand with a call for [logo], but this is also involving NLnet Labs, two universities, Surfnet, and also ICANN and RIPE.

Next slide, please – oh, I have a clicker somewhere.

Thanks. The Root Canary project is also known as the Canary in the Virtual Coalmine. As you know, in the past, folks that worked in coalmines had a canary with them to get an indication of if there was anything going on with carbon monoxide in the coal mine. The canary would simply pass out or die when that would happen. This is actually similar.

For the KSK key rollover, what we want to do is track the operational impact of the KSK key rollover and also get warning signals if something is going wrong there. At the same time, we also want to measure the validation during KSK key rollover from a global perspective to basically learn from this type of event.

So these are the two main goals of the project.

It's basically a measuring methodology that we're developing here, and it's using a couple of different perspectives. The two most important ones are RIPE Atlas and Luminati. RIPE Atlas is the well-known sensor network that you may have heard of. It has about 9,000 nodes spread all over the planet, mostly hooked up to networks of people at home, for example. We're also using the Luminati network, which is proxy network that people use to hide their IP addresses. We're talking to APNIC to perhaps also use their measurements in our methodology.

Finally, we're also considering to use what was called here the offline perspective, doing measurements of root server traffic in an offline way after the rollover took place.

So in essence, this is basically a measurement project.

Also, on the measurement methodology, we signed several domain names, both in a valid way and a bogus way, so to speak. What we're trying to get from this information is the number of resolvers that validate correctly, those that validate incorrectly that produce SERVFAIL, and resolvers that do not validate. As a side effect of the project, we're also measuring which validators support which types of algorithms.

The Luminati network I talked about is an https proxy service. What's important here is that it provides a completely different perspective on the Internet than it does for RIPE Atlas. This covers 15,000 autonomous systems, but 14,000 of those are not covered by RIPE Atlas because RIPE Atlas is usually run by folks who have a networking background and like measurements, for example. These nodes are actually sitting in the networks of regular consumers, so they might actually be more representative of how things are doing than what we're seeing from the RIPE Atlas network.

We ran a couple of measurements around September 19th, which is when the size of the DNSKEY records increased. As you can see here, nothing really special happened. This is the failed responses – the SERVFAILs. Well, there's nothing much changing in this period.

UNIDENTIFIED MALE: So KSK is [inaudible].

CRISTIAN HESSELMAN: Yeah. It should say KSK, by the way. That's a typo down there.

UNIDENTFIED MALE: What's the difference between the red and the blue [inaudible]?

CRISTIAN HESSELMAN: Its history, yeah. And this is similar graph showing the use of TCP and UDP during that period. If something strange would have been going on, then you would expect that the resolvers would have switched to TCP because of the increase of the message size. As you can see here, it didn't really happen, so that looks pretty stable, too.

This is the fragmentations bits. There's not a lot of changes in fragmentation, either. So nothing really happened yet because, of course, the KSK key rollover got postponed. So it's kind of boring.

Anyway, we want to generalize this methodology that we set up now to also measure key rollovers at the TLD level, for example. So we want to apply this more generically and also get more information on how resolvers are behaving in terms of DNSSEC algorithms, for example, but also the types of resolver implementations. But as I said first, we need to have the real deal there.

If you're into network measurements and you're running your own machines, then we would appreciate your collaboration in this effort by running a small shell script that we wrote. It basically uses your default resolvers to query the root servers every hour. If you're interested in joining this effort, then please come and contact me afterwards.

Here's a bunch of pointers to more information about the Root Canary project.

That was my last slide. Russ?

RUSS MUNDY: Thank you, Cristian. Could we have a nice round of applause? Thank you. Let's make sure everybody's awake and change the order of things a little bit here.

Do we have questions for Cristian on the Root Canary?

Roy?

ROY ARENDS: Hi, Cristian. This is Roy Arends. How are you doing, buddy?

CRISTIAN HESSELMAN: Hi, Roy.

ROY ARENDS: You mentioned and then you corrected it, but I just didn't hear it right. I think it's indeed a new ZSK coming in on the 19th of September. Is this correct? Because I think you're measuring the increase in –

CRISTIAN HESSELMAN: Yes, the increase in the message.

ROY ARENDS: Yeah. That's when Verisign, the root zone management partner, does the standard ZSK rollover. They do that every three months. And [they've done this for] years. So I think ZSK is right.

CRISTIAN HESSELMAN: Oh, okay. Then I got it wrong.

RUSS MUNDY: Okay. Do we – oh. Yes?

UNIDENTIFIED MALE: Hi, Cristian. [Yuram] CZNIC. With regards to your algorithm checking possibility in Root Canary, do you have some data from this? Particularly maybe per country – how much some algorithms are supported, something alternative to what Geoff is doing.

CRISTIAN HESSELMAN: Yeah. Actually, we do have quite a bit of data on that, but I did not include that in the slides. But it's on the Root Canary website, rootcanary.org. And there's interactive graphs there that you can check out.

UNIDENTIFIED MALE: Okay. I will check. Thank you.

CRISTIAN HESSELMAN: Sure.

JAPP AKKERHUIS: You can actually see the real-time measurements happening. It takes a lot of computing time, so don't do it too [lightly].

UNIDENTIFIED MALE: I see on the website there are RIPE Atlas measurements. Does it include also that Luminati data?

CRISTIAN HESSELMAN: That's a good question. I'm not sure, to be honest. I need to check that. It should, but let me check that.

RUSS MUNDY: Okay. No more – oh. Oh, yes. Jaap [inaudible].

JAPP AKKERHUIS: For people who want to run their own Atlas probe, I've got five [with me]. So come and see me and I might get one of them. We really need some Atlas probes in this region, so that might be nice.

RUSS MUNDY: Okay. Thank you very much, Cristian. Our next presentation is from Geoff Huston. Can we have the clicker, please?

UNIDENTIFIED MALE: Click, click.

RUSS MUNDY: Click, click. There you go. Over to you, Geoff.

GEOFF HUSTON: Thank you, and good morning, everyone. I'm waiting for the first slide. Wrong slide.

UNIDENTIFIED MALE: That's me.

GEOFF HUSTON: That's you. That's not me.

JAAP AKKERHUIS: You're not Roy?

GEOFF HUSTON: I'm not Roy. Now, I sent these in some time ago, and they could have got moldy.

Okay. Thanks. We've got a bit of clipping going on, but hopefully it's not going to affect things too much. Have I got control now? I do. Okay. Give me a bit to the left. Thanks.

The overall question for ICANN in managing the roll of the root KSK – the real problem is: what number of users are at risk? We know pretty well that approximately, these days, around 11-12% of users perform DNSSEC validation and will not resolve a name it is in valid. In other words, they not only perform DNSSEC validation, but that's all they do. They have no records to a non-validating resolver. So a large number of users, 11-12%, could potentially be affected by a KSK roll were it to go wrong in some way. So that’s the upper bound. That’s what we’ve been able to measure.

Now, there are two risk elements in this roll. The first one is that this is the first time when a relatively large response is an integral part of bringing up a validating resolver because, at this point, the largest response we get at this phase of the key roll is 1,440 octets.

The good news is that that is kind of less than 1,500, which is the de facto MTU size of the IPv4 Internet. The bad news is that not only did v6 pick an arbitrary number of 1,280 – if you ask why 1,280 (the question is insanely weird), the answer is: that’s 1,024 plus 256. What does that mean? Nothing.

Anyway, this number is greater than 1,280. There is an amount of v6 that, oddly enough, insists on 1,280, which is bizarre. But they do. That’s greater. So there is an issue around fragmentation handling in v6 that enters the room at this point.

The second thing you just heard is this whole issue about RFC 5011. We cannot test production. We can test, if you will, toy environments. The only way you can test your production system is with the real thing. There’s no other way to test it on your production environment.

Now, if you fail, you fail [DAC]. The resolver outcome is the same. If you can't get the trust keys, you can't do anything. And it's a loss of service. It's not just where you're serving a zone that isn't signed and you're okay. No. The resolver will just simply not serve anymore. So if the user passes queries only to those resolvers, it'll be a loss of service, which, as I said, is around 11-12% of all users. That's the potential.

We've heard so far about measuring resolvers. That's the question. Users – and what you've heard is resolvers. The way it works is that resolvers that support the signal mechanism send in a signal. Duane explained this.

But let's go into this a little more. Only the root servers get to see that signal and some forwarding recursive resolvers. Nobody else. So it's a signal that I wouldn't say is a secret but is very difficult to find unless you run a root server and look.

Secondly, it's a query without attribution. If you are my forwarder, it looks as if you're reporting, not me, because I'm now invisible. So all those resolvers that use forwarders aren't visible in this. They confuse the attribution of that signal.

Now, you've just heard that a signal is a signal, but in the DNS world, some resolvers are much, much more important than others. Google's public DNS service provides the DNS for around 14-15% of the world's population. Their resolvers are very, very important. My resolver at home supplies answers for me. I'd hesitate to say it's unimportant. I like it, but the rest of you really don't care, nor should you. So trying to understand that some signals are really important and others aren't because the number of users isn't apparent in that data.

Also, what's not clear is that, if a resolver fails, most folk in their [inaudible].com have multiple resolvers. So even if you get [DAC] from A, you might get an answer from B. So it's not as bad as it sounds. If there's a way through, users will find it. 8145 can't tell you that.

So this whole issue as measuring resolvers as an end in and of itself is no useful for this question. It's a wrong answer. What you actually want is an answer that tells you the data you're trying to get. What kind of query would reveal the state of the keys of the resolvers that they are using – and that's plural – back to the user?

Can I figure out if I'm going to be a victim from my DNS service? Will all of my resolvers that I'm currently relying on go black, or will one of them keep on working? Because I only need one and I'm okay.

So can we devise such a query that could reveal that? No. We just can't. Now, if someone says, "You're wrong, Geoff. I have a way," thank God. Love you dearly. Let's go and do it. But so far, a lot of head scratching has gone on. We just can't do this.

Something has to change. Either you create a golden domain inside the DNS that's resolved differently, or you change resolver behavior. We've got to do something.

What if we could change resolver behavior? You go, "You can't do that." We just did. RFC8145 changed resolver behavior. So we're seeing now that the resolver vendors – thank you, cz.nic, thank you, ISC, thank you, Unbound – are prone to changing the code with the right idea.

What if we could? This is the essence of the idea. Can I put in a magic label? It's not a domain name; it's just a label. If you see this label in a domain name and you're a validating resolver, you might want to change good to bad in your answer if – now the first case is this: "Is this key tag a trusted TA key?" If it's not, you were going to send back an answer. Make it SERVFAIL; RCODE3, as I recall. In other words, send back failure if you don't trust that key.

Because we need to detect – and you'll see why – the difference between resolvers that support his mechanism and those that don't, you also do the logical inverse: "Is this not a trusted key?" Send back good if it's not trusted. Send back SERVFAIL if it is trusted.

Three queries. You're favorite domain – because you can do this query as well as anybody else, you can do the query even as a user. Label.some.signeddomain. It doesn't matter what the signed domain. Also, because it's kind of important, set up a label that's deliberately badly signed. I'm sure when you're setting up DNSSEC you've generated a whole heap of those because it's really easy to set up a badly signed one.

If you look at the kind of answers you get, there are four kinds of resolver behaviors that you're interested in. The resolver supports the new mechanism and has loaded the new KSK. I'll get back an A record for the first query, SERVFAIL for the second, and SERVFAIL for the third. The resolver supports the mechanism but hasn't yet loaded the key. The first two queries will switch over. SERVFAIL and A will come in two [as] different cases. If the resolver hasn't learned about this new mechanism, I'll get a different pattern: A record, A record, SERVFAIL. And if the resolver isn't validating, I'll get back all A records because it's not validating. So it'll always work. So I can distinguish the four cases I'm really interested in.

But you don't just have one resolver. You tend to have money, and they tend to be forwarders, etc. The situation is more complex. But oddly enough, the analysis of the results is pretty similar. If I get back from all of your recursive resolvers an answer for the first query, you're okay. If I get SERVFAIL for the second – A, SERVFAIL, SERVFAIL says you're good. SERVFAIL, A, SERVFAIL says you're in a bad place. It's not going to work for you.

The other two kinds of answers say, in all of the resolvers that you use, some of them don't support this mechanism [and] can't tell what the answer will be. That's the two results where you've got unknown. Of course, if you get all As, whatever answer, you're cool because at least one of your resolvers doesn't validate.

So put it in a webpage. Do it yourself. Test yourself. Or, if you happen to know of an online ad campaign – and I do – put it in the Java script or the HTLM5 in an ad campaign and measure a few hundred million users because this technique will actually show no only the state of the key roll but the state of the uptake of the mechanism itself per user. So it's actually possible in an online ad campaign to actually track not only the extent to which this mechanism gets implemented but what it reveals about the state of the trusted keys as this happens.

We're playing with the DNS, so it's definitely necessary to think about privacy and security. This does not reveal end user identities. It's only resolvers, not end users. It doesn't contain any end user identifying information. It need not because it's your set you're setting up, so what you do is your business. You don't have to identity users.

Never, ever change insecure to secure. That's a really bad idea, and this doesn't do that. All it does is change authenticated to insecure under very certain cases depending on your trust key case. So I'm basically turning good to bad, which seems a little safer than turning bad to good.

Anyone can do this. It's not an exclusive thing that root servers or anyone can do. Anyone can do it if you want, and you get the data back yourself. No one else. So the results come back to you as the answer to those DNS queries. It's a more democratic way of doing this test and certainly far more open. ISPs can test themselves. Users can test their ISPs. It's just up to you. It's a much clearer way of doing it.

There is a draft out there. As usual with drafts, at this point the idea is relatively recent. I think it's about two weeks old. If the DNS is your game – I'd certainly appreciate any feedback, and so would my two co-authors, Joe Adamas and Warren Kumari. We are very, very interested in comments you might have about the viability of this approach because, I think, unlike the other one, where you take the RFC 8145 signal, interpret what you are hearing in terms of attribution, and then understand somehow the importance of that signal versus the number of users, this is a completely different approach that tries to look at the state of trusted keys for users and the holistic DNS environment itself. It will not reveal the capability of an individual resolver in general because that DNS isn't like that. It will reveal whether the user is in a good place or bad place about the key roll.

I think I'm now out of time, but there's 46 second for questions. Thank you.

ROBERT MARTIN-LEGENE: Hi. This is Robert Martin-Legene from PCH. I always like your presentations. Have you considered looking at something like the EDNS [inaudible] option for these [crows] hit your name servers to see if the forwarder in front puts those in or not?

Also, one thing that could have maybe detected, if there was a forwarder being used, was if the RFC had included something like putting in EDNS option that would not get forwarded by a forwarder.

GEOFF HUSTON: I've flipped all this argument around. It has nothing to do with the query and everything to do with the answer. This is, if you will, the essential part of this. Almost no information comes to the authoritative name server, and caching doesn't matter. Forwarding doesn't matter. What you are after as a user is one of those patterns from three queries. So it's the answers that come back to you that are important.

Oddly enough, the authoritative name servers in the roots are ignorant. Certainly, they know their own key state, but they don't know the resolvers that you have. Only you know that. This test is for you, not them.

Oddly enough, for Roy over here, in the office of the CTO – Roy was looking after the key roll – they're really after where the headlines in the newspaper are going to happen. That's a user problem, not a resolver problem. So I'm trying to focus this back on the resolvers. So it's the answers that are important, not the queries.

ROBERT MARTIN-LEGENE: Yeah. This specific one is your focus, but the Internet is not only users. Today, there's a lot of automated systems. If something breaks and people don't notice for two months or whatever, that's very likely to happen in automated systems because many systems don't generate any kind of visible errors to anyone. My mom, if her computer doesn't work, figures out to call someone. She's not going to die because her Internet doesn't work. It's different things that need to be measured.

GEOFF HUSTON: What this is susceptible to, if I've got it here somewhere, is actually a large amount of queries – where'd I put about the ad campaign? If we do this right, we can do dramatically large amounts of sampling. We won't test ever user. I'm not sure we can do that. We won't test every device. I'm not sure we can do that. But by the same mechanism that tells us that around 11-12% of users use only DNSSEC validation, we can come up with the same answers here and the same level of confidence about whether a KSK roll will work. That's the entire intent here.

ROY ARENDS: Can I respond to that? I just want to point out that automated systems – people who build them, people who work with them – can also use this technique.

RUSS MUNDY: I think we need to move on now. We want to give Roy enough time to cover his presentation and take questions and answers on the root key rollover state itself. Over to you, Roy.

ROY ARENDS: This is the seventh time this week I gave the exact same presentation, so I would like to see a show of hands of who has not seen this presentation yet.

Good enough for me. I'll continue. My name is Roy Arends. I work in the office of the CTO. I'm a researcher. Small primer – I'm not sure if this is necessary here. If you do DNSSEC validation, you need a trust anchor. A trust anchor is a public key. A public key shouldn't live forever, so you need to renew them. In DNSSEC, we call that rolling the key. You can do this automatically or manually, but there's no way for us – my colleagues over here already mentioned this – to check if you have the right key configured.

When we designed this a couple of years ago, there was a proper design team with [blocks] outreach, etc. We created plans. We talked to vendors. We talked to governments, etc., that October 11th, 2017, was an important date. That's because we actually didn't know who was validating or not or who had the right key configured.

Naturally, we have a process for this. On July 11, 2017, we introduced a new KSK, and then we monitored it for any fundamental changes in root server traffic. My good friend Duane works for Verisign, a root zone management partner. He had access to A and J root zone traffic. We have access to B, D, F, and L root zone traffic

I need some water.

UNIDENTIFIED MALE: [inaudible] some water.

ROY ARENDS: No. I got water. We got access to B, D, F, and L root traffic. Eventually we got all the root servers to provide [inaudible] to us, which is nothing else in checking how often DNSKEY queries are sent. You all know that, if a resolver [can't] validate, it gets very aggressive. It will just go on asking.

A couple of years ago, Geoff and I did some research on rollover and die. We can call it now rollover and comatose, right? Because a similar effect is still in place. Validating resolvers will go out of their way to get the DNSKEY. But nothing happens. We've got beautiful stats on that, beautiful graphs, but it just shows a kind of flat line on July the 11th.

30 days later – this is a 30-day hold-down period. Duane had a beautiful graph in there, where you can see this fantastic swap – there was still [inaudible] traffic. Nothing to see there. So we continued.

September 19th. This is what I alluded to earlier: when our root zone management partner, Verisign, introduced a new ZSK. This is nothing special. They do that every three months. This is standard ZSK rollover. They've been doing this since 2010. The only exception here is that this is the first time the size increased. The response, I think, [increased]. We talked about that as well, so I'm going to continue.

You've seen this slide. It's from the [inaudible] presentation. Until recently, we had no knowledge on who had what trust anchor configured. We already talked about RFC 8145. We already talked about BIND9 and Unbound, so I'm going to skip this.

It says here, "No other known implementation." I just learned from [Andre Phillip] that KNOT will have in the near future a version with the [inaudible] in place, so that's good. I don't know about PowerDNS recurser. If they have it, I don't know. But Microsoft recurser –

I've got a number in here. About 4.2 million unique addresses sending queries to root servers. This is an old number from B-root servers. Forget that number, if you will. It's probably a lot higher if you combine all the root servers together. If you do a longitudinal study about the number of unique addresses, naturally the number goes up. So 4.2 million is lower bound.

The numbers we actually got with RFC 8145 data is very, very low. I'll go to that in a second.

Here we are. These are statistics we have. It's until October 24th from the 1st of September. This is when I had to fly to the RIPE meeting. We have about 27,000 out of those 4.2 million reporting data. The total number that only ever report KSK-2010 is 1,631. That's 6%.

Now, I did not look at if these resolvers are actually validating, so they might not have the [DO bits set]. They might have the [RD bits], someone basically checking if this key tag signaling thing actually works. If you do that, the number will go down slightly, but not that much.

My friends already alluded to this. The analysis is complicated. I actually really like Geoff Huston's proposal. I wish I had that two years ago. I really like that – oh, before I say something wrong, that's a personal opinion. I'm not speaking for ICANN. But I really like the proposal.

We know of a few reasons why KSK-2010 is reported and not KSK-2017. The latest BIND version reports the trust anchor, even though they're not validating, which is annoying because now you have this false signal. You're not doing validation. You have this key configured and you're probably not going to update because you're not validating.

Originally, before 5011, there was this old configuration [inaudible] in BIND that says trusted keys. There you put your trust anchor, which makes sense. But then we got RFC5011, and you need a different configuration to [inaudible] now and to make sure that some keys can be updated with 5011. The others are manually configured. So BIND uses managed keys to do 5011 managed keys. Some people are confused about that.

I'll skip over this. There are many issues. There are too many to report here.

Back to the plan and process, I already said that, on the 19th of September, there was nothing going on here. Around that time – this is right before the DNS-OARC meeting – we got Verisign's report, and we corroborated with our own data.

We can't just decide things ad hoc, so we consulted the operational plan – this is a plan that's a few years old; the community has ratified it – and it say the root zone management partners, Verisign and ICANN, might also to extend the new phase for additional quarters. For example, if new information indicates that the next phase may lead to complications, the current phase would be prolonged. This is referred to an as extend scenario. On the 27th of September, the extend scenario kicked in. The moment we decided this, we made it public.

I've heard some reports that some mailing lists got this information earlier than others, but that is because our communications team are not on all DNSSEC mailing lists. So we had to forward that information really quick.

Like I said before, we don't know how representative the set of validators reporting key tag data is. Geoff already mentioned that validators are not the same as end users. I'm planning to work closely with Geoff. We can share some data, Geoff and I, so the numbers that are reporting these auth key tags he can actually ballpark, if that makes any sense.

The number of users behind it? I heard a number the other day that 750 million are currently behind validating resolvers. Is that correct?

GEOFF HUSTON: Google.

ROY ARENDS: Behind Google. Sorry. Okay. Mitigation is hard. We already had another year campaign to reach operators. These implementation-specific problems that I just mentioned before don't make the problem easier. We've all written software in the past. We all know bugs exist. We can only hash them out when you actually do this. Like Geoff said before, you can only do this roll operationally. We can't test this in a lab environment.

We postponed the key roll until we get more information and understand the situation better. It will be at least one quarter. For those who have questions about that, that means one quarter or more quarters. It's always aligned on a quarter. The reason for that is the key ceremonies that ICANN organizes every three months. We have not yet determined how many quarters to delay.

We will at least partially mitigate. We've hired someone, a contractor, to track down the initial 500 that we saw in September. This was before Unbound was released with the key tag signal on by default.

Data collection continues. I just want to say one thing about Unbound. Duane mentioned that Unbound was released on October 10th or 11th and that that coincided with a signal. There's not a day that coincides with that signal. It's that the 10th of October is also the last day that ZSK-q3 was used. Part of the uptick was that ZSK-q3 was added to the signal.

Also, just because Unbound is sending that query does not actually mean that Unbound is validating. It might be a f0warder in front of a –

JAAP AKKERHUIS: No. It only gives a signal when it's validating. It's different than BIND.

ROY ARENDS: Okay. If I ask Unbound a query that uses this key tag, which is a top-level domain, basically, will it go to the root and ask it? It resolves the key tag.

JAAP AKKERHUIS: It really depends on how you configure to complete to the DNS chain. It's a caching resolver. If the user has put in some way to [inaudible] forwarders – people do most weird stuff, like putting in forwarders. Some of them might actually –

ROY ARENDS: Sorry. You're eating into my time. Let's take this in a minute. I'll show you some data later. I was just helping you hear that Unbound is not always a problem.

Anyway, the last thing I want to say about this part is: please do not remove KSK-2017. What happened when we announced that we delayed the roll is that some people decided to unconfigure or remove KSK-2017 from their configuration. Don't do that.

I've got another slide deck really quick. This is about root KSK roll frequency. This was a presentation I was supposed to give at the Technical Experts Group, but I got three minutes for that. Now I look at the clock and I only have one-and-a-half minutes for this one.

Really quick, this is a small discussion about KSK roll frequency. We started to use the old key in 2010. We now plan to start using the new KSK in 2018 if nothing else happens. Some folks are adamant about manually configuring the new trust anchor. Some folks rely on 5011m, and some are using non-5011 automatic update stuff. Validators might have some configuration issues: read—only partitions or using the wrong configuration stanzas. We know that bug exists. Trust anchor telemetry is far from optimal.

There are two schools are thought. Roll frequently. That doesn't mean often. I need to make that clear. Roll frequently doesn't mean roll often. Roll frequently means at a set interval. Roll infrequently means, "Hey, it ain't broke. Nothing happened, so why should we roll?" So it's not with a set interval.

The DNSSEC technical community a couple of years ago went with a kind of roll frequently, which was five years. I know we now [inaudible] that.

I decided to put in here some frequency band, if you will. Frequency band is lower frequency and upper frequency. Upper frequency is easy. Let's take five years for now. The absolute lower bound is three months due to the key signing ceremonies that we have. If we want to change the key more often than three months – I know you might find this ridiculous – I've heard people say we should do this every week. That's why I put this in there. The lower bound, the highest frequency, is three months. Anything higher needs a big, big amount of work.

What's the effect of doing this every three months? We have three stages that deal with introducing these keys. If you do this every three months, you collapse three stages into one. You introduce a new key, you start signing with the current key, and you revoke the old key. That's a big problem. I'll skip the second line. It naturally needs to be redesigned – the entire operational plan – but the minimal DNSKEY response size is 1,986 bites. That's kind of a no-go.

UNIDENTIFIED MALE: [inaudible]

ROY ARENDS: Yeah. Also, manual or semi-automatic deployments will have to update every three months. So I'll just put this out there. This is not an opinion. This is what will happen.

When you do the six months, it doesn't have the response size problem because you don't have to do this complete collapse. You still collapse two stages into one. One stage is that you introduce the new key and the second stage is that you revoke the old key and use the new key.

There's a small problem in there. If you do this collapse of these two stages into one, you cannot roll back. So we need to redesign the entire process. I think it's actually a good idea to have the option to roll back.

Effects of rolling every nine months: it doesn't have the response size problem, you can use the current design of the roll plan, and it's the highest frequency that does not incur fundamental changes to the design. This is from the guys who roll the keys perspective. I noticed different opinions from the operational community, but this is from our perspective.

However, this might still be awkward in regards to timing. Every year, the time slot moves forward a quarter. Every four years, we will have it twice in one year. It's not optimal or operators due to lack of predictability. That's my opinion.

Effects of rolling every year: it doesn't have the response size problem, you can use the current design of the roll plan – all of this good stuff – and it's more predictable and hence likely better for operators.

Effects of rolling after more than one year: no significant difference from doing this every year. It's kind of Groundhog Day if you do this every year or every N years at the exact same time.

I can imagine that, if we do this frequently – that doesn't mean often; it means with a set interval; we do this not faster than one year and we have an upper bound of five years.

I talked to Jaap yesterday and he made a very good point. I don't want to steal the question away from you, but Jaap had a very good point. He looked at me and said, "Should we first do a key roll before we make any decision?" He's absolutely right. I promised to do this presentation. It was forced upon me. So I think we can still have this discussion. But, yeah, it's premature due to that we haven't done a roll yet. Thank you.

RUSS MUNDY: Thank you, Roy.  Duane, go ahead.

DUANE WESSELS: Roy, can you go back to the six-months slide for a second? Can you clarify? You said there's no way to roll back, but what about extend? Is roll back and extend the same thing in this case?

ROY ARENDS: No, it's not the same thing. If you revoke the old key the moment you start to use the new key, you can't roll back. But if you delay revoking the old key for a quarter, you can still roll back.

DUANE WESSELS: Well, I'm thinking of the current situation, where we've extended – the pre-published period, basically. We've extended and not gone to the next step. Maybe it's too hard to figure this out right now, but we should maybe talk more later.

ROY ARENDS: Yeah.

RUSS MUNDY: Yeah. I think we've touched on some very important things. One of the things is that, since we're a group that is associated with SSAC, SSAC63 dealt with a lot of these issues several years ago. One of the important things that was noticed is that we should try to learn as much as we can from this roll so it can be used in future rolls. I think this is exactly what we're starting here. So this is excellent.

Do we have any other quick questions for Roy? Lunch is waiting for us for those that are joining us for lunch.

Robert?

ROBERT MARTIN-LEGENE: Yeah, you can't keep me down. One thing that could be considered in terms of all of this – what works and what doesn't work – would be to have, let's say – you're all going to go think I'm crazy – half the root servers supply data signed with the old key and have the other half signed with the new key. Then you'd see who'd keep asking – stop saying no. You wouldn't have a denial of service, actually, for the user, but you'd see who'd be using which key.

ROY ARENDS: This idea is not a bad idea in and of itself, but it has been mentioned a couple of times and we've looked into this. The idea is basically to have a set of root servers serve the root zone with the current key and not a new key. The idea is then to watch this natural progression of both configured trust anchor validators to go to this new – that's incredibly hard to measure. It really doesn't help because what we really want to do is, at one point, have everyone get over to KSK-2017. Then you have the question: When are we going to switch that off? So it doesn't help. It doesn't give us any new information. We already know who they are. That's what we can see here, assuming that this is a proper sample. So my gut instinct says no.

ROBERT MARTIN-LEGENE: I'm not actually thinking it should be done, but it could be done. I understand very well why ICANN wouldn't want to go that way.

ROY ARENDS: Yes. There are many things that could be done that we want to do.

RUSS MUNDY: I think we want to give Roy a great thanks for doing this presentation for his seventh time. Thank you, Roy.

And now, Jacques – where are you?

UNIDENTIFIED FEMALE: He went to the restroom. He'll be right back.

RUSS MUNDY: Ah. The Great DNS Quiz. What I think we're going to do, since we are over –

UNIDENTIFIED FEMALE: We have time, Russ. Lunch is not until 12:15.

RUSS MUNDY: Oh, it's not until 12:15?

UNIDENTIFIED FEMALE: Yes. We've got a buffer in here, so we have time.

RUSS MUNDY: Oh, good.

GEOFF HUSTON: Can I just make a little bit of a…

RUSS MUNDY: Yeah. Go ahead, Geoff.

GEOFF HUSTON: Part of the reason why no is a really clear answer there, Robert – and he's not even listening – is that, when you have the two keys listed in the DNSKEY resource records signed by the old key, that implicitly allows you to trust material signed by the new key because you trust the old key. Even if you went to a different root server, because you still have got the old thing in your cache, you will trust everything your modified root server is saying because you trusted the old key. And the old key signed across the new key.

So this whole issue is complicated by that old-key-signs-new-key. It means you trust the new key for as long as it's in your cache, signed by the old key. That's why separating it out inside the root with distinguished domain names that have different DNSSEC behavior – "Here's a name that can only be validated with the new key if it’s in your trustor” – requires work in the root, work on the validation protocol, and work in the resolvers. It’s perhaps the hardest way of doing it, and I’m not sure that the data would be useful because, as I said right at the front, it’s not resolvers you have to worry about if you’re rolling the key. It’s the damage you’re going to do to users. You really need to structure your telemetry to get away from a focus on how resolvers work and refocus on users.

Jacques has entered the room, so I need not waste any more time. Thanks.

JACQUES LATOUR: Any other questions?

UNIDENTIFIED MALE: You’re up.

JACQUES LATOUR: All right. Well, I’m not sure how I became the quiz master. That’s because somebody stopped doing it, right?

UNIDENTIFIED MALE: [inaudible]

JACQUES LATOUR: No? Heh. All right. That's right. Welcome to the Great DNS and DNSSEC Quiz for ICANN60. In front of you you should have the quiz with ten questions to answer. So you have the sheet. If you don't have any, you can grab some on the front tables, maybe. Everybody should have a pen or so.

All right. Here's a rule that I learned from Roy: I'm right. Yes, I can be wrong, but it doesn't matter. I'm right. So that's where I'm the authoritative quiz master server.

It's one good answer per question. It's one point per good answer. It's so simple. You have ten questions and a maximum of ten points. There's no way you can go to -17 or -19 or -20.

UNIDENTIFIED MALE: [inaudible]

JACQUES LATOUR: That was good. Let's start. Question 1: What was the date the root KSK rollover was supposed to happen? The KSK is used for assigning the root zone key set. When was it supposed to happen? A) October 11th, 2014, B) 15th, C) 16th, D) 17th, or E) It was supposed to happen last year but we decided not to do it?

Three…two…one…zero.

Question 2 – that's a different one, eh? – Which Twitter feed tracks the root zone change? @therootchange, @changeroot, @diffroot, @root – Jessica, scroll up a little bit because the text is…that's annoying. Or @IAmGroot?

Next [inaudible] Question 3: Which mailing list should you subscribe to? What's the best one to view on report on issues with the DNS in general? DNSoperation@list.DNS-OARC, IAmGroot@list.DNS.DNS-OARC, DNSSECcord@elist.ISOC.org, DNSop@IETF.org, or HelpWithDNS@ICANN.org?

Remember, I'm always right, so whatever I think is the right answer is the good one.

Three…two…one.

Question 4: In the draft IETF homenet.14, what domain is designated for non-use in residential home networks and designates this domain as a special use domain? A) .home, B). [rpo]/home, C). home. [.mezo.kaza.mezo] – I think that's Chinese – home. [rpo] or .mezo?

Question 5: Which ccTLD was most recently signed with the [DSN] root? A) .ax – I can't say it – .[Alain Island] – .gw [inaudible] – .bm (Bermuda), or .sa (Saudi Arabia)? Most recently signed. We talked about that this morning.

Three…two…one.

Here's a special one. It starts with sorry because I'm Canadian. Which one best describes the DNS-based Authentication Name Entity (DANE) TLSA certificate usage value #2? A) unassigned, B) PKXDA-CA – poor translator; he must have a hard time with this – C) DANE-TA, D) DANE-EE, or E) PKIX-EE?

This is so obvious, right? Yeah? [inaudible] Yeah.

All right. Next one. Question 7: What percentage of all TLDs in the root are signed with a [second delegation] APS in the root? We had that slide in this morning when everybody was sleeping and I said, "Ooh, this is important."

Hey, I like this one. Which two papers by Paul Mockapetris published in November – you need to scroll there a little bit – 1983 marked the beginning of DNS? So which two papers? A) BCP16 and BCP17. Is it B) BCP42 and BCP78? C) RFC882 and RFC883, or D) RFC1034 and RFC1035?

That was a Google search right away.

Three…two…one.

Question 9: What is a network addressing and routing method in which [data gram] from a single sender are routed to any one of several destinations selected on the basis of which is nearest, lowest cost, [LTS], with the least congested route or some other distant measure? Is it Multicast, Anycast, Unicast, or Star Trek Subspacecast, or [Flurrycast]?

Oh, that's okay. Three...two...one.

What does DNS stand for? Is it Domain Name Server, Domain Name System, Domain Name Software, Domain Name Service, or Domain Name Space?

The reason I put the question there is because I'm not sure what the answer is.

UNIDENTIFIED MALE: [Then how could you] be right?

JACQUES LATOUR: I'll pick one and I'm right. That's it. I'm not sure we're going to have a discussion over lunch.

DNS – oh, yeah, yeah, yeah. That'd be quicker.

All right. So corrections. Pass your sheet to your number. Super easy. It's one answer per question. It's one point and a maximum of ten points. We're going to find out if I was wrong, right?

Question 1. The answer is D. It was supposed to happen October 11th, 2017.

UNIDENTIFIED MALE: [inaudible]

JACQUES LATOUR: Are we good?

UNIDENTIFIED MALE: We're good.

JACQUES LATOUR: So far so good? That was a [gimme]. You need at least one point to have the lunch ticket. That's our new sponsorship role, right, Cristian? Yeah. If you have zero, you don't go for lunch. Too bad.

Question 2. The answer is @diffroot. IAmGroot – I like that one.

Question 3. A. The DNS operation list is probably the best place if you have issues with the DNS. If you're not on that list, you can go Google it, join, and be part of it. It's a good resource. See? I'm doing a little bit of marketing here.

Question 4. The answer is D. I like C. We should have this huge domain with all the names and all the language, .home. It just works.

Question 5. .gw (Guinea Bissau) was added in October 2017. That was the latest addition.

Question 6. The answer is C. Did you get it right? Yeah. That was just for you. Trust anchor [assertion] is the right answer for DANE.

Question 7: C. Their number is: 90% of TLDs are signed, exactly.

Question 8: That was RFC882 and 883 that were written in '83. Is there a relationship between 883 and '83? No.

Question 9: Anycast. Eventually the next version is Subspacecasting. We're going to start working on that pretty seen. [Flurrycast] is what we'd send everywhere and they all respond and [hope]. I like that.

Question 10. I didn't really know what the answer was, so I went for B.

All right. Now we're going to figure out who the Great DNS Guru is. Count the score. Raise your hand if you have five or more and leave your hand up.

Six and more. Seven and more. Eight and more. Nine and more.

UNIDENTIFIED MALE: [inaudible]

JACQUES LATOUR: Ooh. And ten? So we're in the presence of a Great DNS Guru. Woo-hoo!

I guess we should have a star and just –

RUSS MUNDY: What you definitely get is a free lunch.

JACQUES LATOUR: Yeah. So you get the free lunch. Nine out of ten? Pretty good. Excellent. The lunch is –

UNIDENTIFIED FEMALE: The lunch is in Hall 4. If you just go past Registration, take a right. It's the big, huge room. Just so you know, you don't have to go through the metal detectors. Just enter on the right, and just remember you have to have your ticket.

RUSS MUNDY: Okay, everyone. Please go and –

**[END OF TRANSCRIPTION]**