MARRAKECH – DNSSEC Workshop Part II
Monday, June 24, 2019 – 10:30 to 12:00 WET
ICANN65 | Marrakech, Morocco

UNKNOWN SPEAKER: Good morning, next up is Tim April and he's going to talk about that KSK Roll.

RUSS MUNDY: I want to express particular thanks to Tim for his willingness to do a Geoff Huston presentation which in itself is a challenge. Geoff originally said, "Well, I might be able to do this remotely." And Tim was kind enough to raise his hand and said, "I can do it in the room for you." Be kind to Tim but it is a Geoff presentation. Over to you, Tim, thank you.

TIM APRIL: Thanks, Russ. I'm just flipping through to make sure Geoff isn't watching to see me butcher his talk, I'm sure he'll watch it on the recording. I am not Geoff Huston and I'm not going to try and be as entertaining as he would for this. This is a presentation he gave at RIPE a couple months ago or maybe a month ago, I can never remember. There are two major particles that are used in the internet that are really hard to get good data about, the first one being BGP and the second one

being the DNS. This one is talking about rolling the KSK which is mostly done now for this round. I'm pretty sure the announcement was pulled through revocation a couple months ago.

All of these slides were originally Geoff's, I've modified them very slightly to make it so that it's a little bit more applicable because while all of the statement of I'm not part of -- I'm not part of the ICANN Organization or PTI, while APNIC is not a root server operator, in my day job I am also not a root server operator, I'm not a member of the root server organizations and I cannot see the root server query data, other than parts of the diddle that the day in the life of part of DNSR and I am also not Geoff but I'm presenting because I drew the short straw.

Why is this of interest to me? I am a end user of the internet. I do DNSSEC validation on my machine. I run a resolver that does DNSSEC validation on my own and then I also work a company does a lot of DNS traffic and we do a lot of DNS validation for our end users. Back in 2010 the Root Zone was signed and at the same time there was an agreement between the Root Zone Maintainers and all of the Community about how -- the key would be being able to be rolled. It said that it should be rolled about every five years but that was kind of hand waving and all

of that. The KSK is a special key where there is no parent key for it to be signed by.

Every validating resolver must get this key through some method in order to do validation all the way up to the root. When it came time to rotate it, this was the first time we'd ever rotated the root key so we had to figure it all out from the start. They came up with the approach where the old key is going to sign the new key for some period of time, they're going to publish the new key in the root zone and validating resolvers are supposed to figure out how to add that new key to their resolution path, whether it be through RFC5011, where they notice it in the root zone for the set period of time and then they add that to their trust or whether they go and change the code or things like that and publish the new key.

The original plan from ICANN was as you can see in this plot, where in -- they would start by publishing both the old key and the new key, where the new key is signed by the old key and then they'd start signing the ZSK, the Zone Signing Key, the key that's signed by the root KSK with both keys at the same time and then they'd switch to signing with the new key. This was supposed to happen -- the plan was for the root zone signing to switch on October 11th of 2017. Then all the technical sessions started and ICANN decided to postpone that key rollover and at

that time it was indefinitely but it turned out that they were going to postpone it exactly one year to October 11th of 2018. This had a convenient artifact where they didn't remove the key from the root zone at the time, so we had a full extra year of the new key being in the root, which could prove to be interesting when we try to roll the key again if we don't have that extra year of key being there, we may see different results or we're likely to see different results anyway.

Now that we've gotten to the part where the key has been rolled, so if you're doing DNSSEC validation you're using KSK2017, what do we do with the old key? There was a discussion in the KSK Rollover Mailing List of what we do about it and since no one expressed any strong reservations about getting rid of it, the key has I think in the Virginia Fifth Key Management Facility, that key has now been destroyed, that was a three hour livestream of Matt Larson with a screwdriver taking apart AHMS, it's all riveting material that you can fast-forward through on YouTube if you really want to go watch it. The next operation that's going to happen is August 14th at the West Management Facility, where they're going to destroy the other copy of the key. If you have any strong interest that we should keep that key, speak up now or it's going to go away very soon.

What worked this rollover? As you can see here this is the plot from APNIC of the DNSSEC validations that they see doing their testing, where there's not a significant dip on the 11th. It looks like most validating resolvers didn't have a problem. If you look further into the data, it wasn't incident free, so there were some impact end users, going into it, it was really hard to predict how bad the impact would be on the entire world. Measuring the DNS can be tricky, especially because it's not always implemented in the same ways everywhere.

One of the big questions that I remember hearing back in September of 2017 was, how much impact to the end users is the ICANN Community willing to accept and can we measure what we think the impact will be? It turns out that's really hard to do, especially because the protocols weren't designed in a way that make that apparent to anyone unless you're the Admin of a specific machine can you tell if it's going to work for you or not.

There were a couple methods of trying to indicate this or signal this data. The first was, we can signal to the root name servers for example, where if the DNS resolvers send a signal up to the root name servers of whether or not they have what keys they are trusting, we can then ask the root name servers to go and look at that data and see what they can figure out of how much

impact there will be. This was RFC8145 signaling where the resolver will tell the root name servers which local trusted keys it has but that had some tricky pieces to it of this is what the root name servers were seeing from that data, where once KSK 2017 was published there was no significant drop in who was accepting the new key until the RFC5011 window hold down timer ended, when you can suddenly see a large majority of resolvers started reporting that they were trusting both keys. There's still that line down at the bottom, that red line of not updated resolvers and the scary part is where it starts going back up towards the end of October.

If we look at the data in another view you can see that by about the time the key would have rolled the first time, there are about five percent of resolvers that were reporting that they didn't have the new key or they didn't trust the new key. That would be one way of measuring, that would be five percent of the internet that would have DNSSEC validation failures.

Once we waited a few more months, you can see that in about, I think that's May of this year, the same signaling data shows that about one percent of the internet would have resolution problems. The problem here is that's mostly an artificial way of measuring the internet. If users don't uniformly use different resolvers around the internet, so if it was a resolver that only one

person is sitting behind and it doesn't have the new key, it may still be signaling to the root name servers that it's using KSK 2010 and that's going to only affect one person and that maybe one of their secondary resolvers, so if they get an X domain from that machine, they may ask another one and they may still be fine, that may not have any impact at all.

If you chain resolvers behind each other, if there's one resolver all the way down in someone's home router, it may be going to another resolver but it may be sending that trust anchor query up through the other resolver, so it may make it look like that resolver doesn't have the right key and you may assess the impact wrong again. There's also the tricky part that some queries just keep going around the internet without any known end of when. You may send one query years ago and it may still be repeating forever.

There're all sorts of issues where if you got chained resolvers, you end up seeing data where one resolver maybe looking like it's not updated but it actually is and it's one of the downstream resolvers. If there are two resolvers that are behind one resolver, you may not be getting enough of the queries to tell you how many resolvers are not actually updated because of caching and then there's no good way to measure the outcome

because you can't determine how many users are behind a signal resolver based off of that sort of pattern.

Another way to test is by using the client to do the testing. If you have the ability to run code on a whole bunch of clients around the world, either through Ripe Atlas Nodes or through the method that APNIC uses for testing internet, you could send queries to the client, where does the KSK sentinel query that we're published in an RFC that I can't remember the number off hand, where if you send a query for the root key sentinel is TA and then key tag to a resolver that supports this RFC, it would return with either a serv fail or a -- I can't remember if it's a wreck, Warren can correct me on that if I'm wrong and then you can also send the opposite of Is Not TA and then the key tag and it will provide the opposite response. With the correlation of both of those and the responses that you get back, you can determine whether or not the resolver supports the RFC and then whether or not the key is rolled or not. That wasn't widely deployed so we couldn't get a good measurement through that because that RFC published, I think shortly after the key roll date. You can that testing through just a webpage or through embedded ads and things like that.

Prior to the KSK roll there were about 16 percent of resolvers were validating, I think we're now up to 19. Going up to the roll,

there was the expectation that about point one to point two percent of the users would end up having DNSSEC validation failures but these measurements were very uncertain as I mentioned for both methods. When the key roll actually happened, this is a very nice plot that I know a bunch of people were watching the day that the roll happened, you can see the blue line is the resolutions that were scene using the old key and then the green is the new key. I think it was just about 24 hours, I guess it's 48 hours.

Then we get on to the point during the roll we didn't see a significant drop in validation and people reporting that they had the good or the bad keys but other people had different things to say. There were a couple reports that came out that there were ISP's that just stopped resolving properly. This is one new story about an ISP that all their resolution stopped working, I believe they fixed it shortly thereafter. If you're an end user that is behind the validating resolver that didn't update, you would just black, there was nothing, you internet just stops working and there's no good way for an end user to figure that out expect call your ISP and that's when some ISP's started to respond. In the case of the one that the article came out, you can see that there was a significant drop in the sample data from APNIC of how many queries they actually ran, where the KSK roll happened and then all of sudden they stopped doing anything.

Also, in the data at the same time, there was a significant drop across the rest of APNIC's testing data, it's hard to say whether that was KSK related or if it was just natural traffic patterns.

Looking further into, Geoff took all of the ISP's that he had that had more than 400 samples per day where there were 30 percent of validation going on prior to the KSK roll and then checked to see where there was more than a one third drop in validation after the roll and came up with this list of ISP's that he believes turned off some sort of validation right at the time of the roll over.

There were at least three networks that disabled it and left validation off and they're highlighted here. From the view I was looking at, that's not as many as I expected to have happen but still not a great impact.

It appears that the end user impact was about point two to point three percent of users based off of this measurement. 32 ISP's have restored validation since and three still have it turned off.

That's not the complete story because the next event was the revocation. On January 11th of this year the revocation bit was set, it was supposed to be an easy thing where they just add the revocation bit to the signature of that key but then the result of it -- revocation for end users no significant change but the root

servers saw a different thing. Once the rollover happened, there was a slight increase in DNS key queries towards the root, this was reported by an A and J root. After the revocation, the queries started to explode. I don't know what happened after the key was pulled from the zone file a couple months back, maybe Warren or Wes have more insight into that.

Digging a little bit more into it, at the time it was believed that something like bind was having an issue. Bind sent some mail to the KSK rollover list saying that they believed they found the source of some of that traffic. There was a loop where in a specific condition bind would send extraneous DNS key queries to the roots, there are still some discussions whether or not that was actually the issue.

The lessons learned, the fact that we can roll a KSK. This may have been a special case because there was quite a bit of publicity about it, there was a lot of talks, we had an extra year. It's still unclear whether or not we can do it again in such a clean way. To make that harder, DNS is really hard and really hard to measure how the DNS is working across the network.

This roll was a good experiment. The trust signaling could use some work. There's a lot noise, it's not a great, clean data stream like everyone would hope for it to be but nothing ever is. It would be helpful to try and make some more effort to make

this more observable to the operators of the network, to have a better idea of how much impact there will be in future rounds of the roll.

DNS validation is most appropriately a resolver function and not an edge function. Resolvers are the ones that we really have to care about here. The importance of the resolver is partially related to the number of users behind it. We have to question whether or not 5011 was the best way to communicate the key to everyone.

This is from Geoff. Geoff's view of should we perform this sort of thing more often? Should we continue to roll the key? How would we continue -- if we decide to keep rolling a key fairly often, how do we communicate that key out to everyone that needs to get it? Should we consider rolling the key algorithm? That's going to be a whole other level of difficulty.

There was a long discussion on the KSK rollover list of whether or not there should be a backup KSK provision in the root and then we just roll from the current one to the backup and then create a new backup and just keep that rolling method going?

Another view is, why are we rolling the KSK, it was never compromised or we don't believe it was compromised? It may be more trouble than it's worth rotating it without any issues

like that. It the key were to be compromised, with this model we'd have to wait 30 days to roll the key anyway, so this isn't a great example of how we would respond in a major incident like this.

Is doing this again going to teach anything new?

Is old signing new really the best way to do this?

How should we scale the KSK going forward?

That's all I have. Are there any questions about it? I can try and answer them or some of the people in the room who are more involved can.

RUSS MUNDY: I'd love to hear folks jump up and ask some questions. I watched the video of Geoff giving this and drew some questions at RIPE. We've had a number of presentations here in the workshop about the KSK roll and I'm very appreciative of Tim doing this presentation from Geoff but we don't seem to have a lot of people heading to the mic. Let me ask Tim if he has any comments on especially, they two alternative views expressed at the end, which basically are do it a lot or why are we rolling at all?

TIM APRIL: I made the comment to Geoff I think a while ago about this of I think we came from different sides of the roll or not argument. I believe he was on the more hesitant to roll but I'm firmly of the opinion that we should crawl it fairly frequently. I sent mail to the KSK rollover list of, I would propose we move forward with the backup model where either every year or every two years we roll from the production key to the backup key, provision new backup key at the same time and just keep that rolling model, that way if for some reason there's a compromise to the HSM or the process or something like that, we can immediately roll to the new one, that doesn't take into account if there's a compromise of the process, that means both the production and the backup key are compromised, that's something to consider. If we wait another five years, we're just going to get -- ideally, we get to something like 50 percent validation and then it gets more interesting to ask whether or not we can roll safely or not. The operators will get complacent if we don't keep rolling it on them.

MICHAEL: I'm in favor of frequent key rolls as well as perhaps also including an EC key as I think RSA is starting to reach its end of life in the next five to 10 years as performative practical security. My biggest concern with more frequent rollovers at this point is we don't have a good infrastructure of deploying new KSK keys

to devices because a lot embedded IoT devices or home routers that do validation, they generally only need a firmware update to get the new KSK and if we're going to roll frequently we need a good infrastructure in place to provision down the new key from the root zone and make it all happen magically and transparently in the near future.

TIM APRIL: I forgot to mention that when I was talking about RFC5011 of IoT and devices that are firmware based that can't write to the -- that have no permanent storage ability, that have no ability to write to their permanent storage without a firmware update can't really take advantage RFC5011 because every time they restart they're just going to switch back to the old key and that gets tricky for helping end user, CPE devices to update going forward.

MICHAEL: If we had a relatively simple way where the key can be pulled from the root zone via standard DNS query, it wouldn't be hard to design IoT devices because most have at least a couple megabytes of -- I'm talking as a device designer here because I've built more than one of these things, where the most recent key could be stored. It's just that the embedded operating

system such as Broadcom Wicked or ECOS doesn't have any provisions in its software stack to grab new key and rebase how it does DNSSEC validation. If we had a standardized way of pulling the key from the root zone when the new key kicks in when the new key goes away so forth and so on, it would be possible to bring this to the IoT world and not require a firmware update.

TIM APRIL: There is a way to get the key from the root zone because it's published in there.

MICHAEL: Yes, but it's not standardized, you can validate the zone with the old file and then get the new key that way but I don't think there's a fact a way of saying which keys should be live versus not live. For example, in Ubuntu we had to push a stable release update, add KSK2017 to all releases of Ubuntu [inaudible] keys mechanisms; that's how we did it. I'm pretty sure Red Hat and everyone else did something pretty much identical to that.

TIM APRIL: Warren may have a better answer then I do, we can talk afterwards.

RAMANOU BIAOU: Could you go back to the previous slide? On the why are we rolling the KSK? Seems like there's a conflation of a couple of things. There's one, risk of process. There's an instability risk where you have a process of rolling it and if it goes wrong there might be a problem but that doesn't that have to be compared with an actual real life scenario where there is a compromise and you don't have a process that has actual been worked out and then you have a real problem situation? I find it very hard to be sympathetic to, it's not broken, let's have this leap of faith that it's going to be okay and we're intentionally introducing instability but it's actually if you don't understand how to roll it and if you don't have operators understanding how to handle it, it's not just an instability problem at the point, then you have a security problem. It seems that that point ought to be clearly made across the board.

TIM APRIL: Yeah and there was also -- as I watching the key destruction operation happen, they had a significant issue where they thought that -- I had to step out part way through but they were having trouble booting one of the HSM's to point where they thought that the HSM had zeroed itself already, that made me really start to think that the backup key would be important,

especially on alternate hardware so that if we find out that the HSM's have five year shelf life and we keep the key for seven years, well we're going to be in a real problem soon.

WARREN KUMARI: I'll point that Paul Hoffman would beat me if I don't mention the KSK-Rolloever@ICANN.ORG mailings list, if people are interested in this topic please join that and comment on that, that's where a lot of this is being described or discussed. Kind of falling on from what Ram said, there's are sort of two reasons or two primary reasons for rolling the key. One of them is just standard DNSSEC hygiene, crypto hygiene, make sure that you know how to do this. This key probably is safe for a while but it's always unclear what safe means, in 10 years' time is it still good? There's always the specter of quantum, etc. This process, while it has a bunch of what's can kind of be used to solve those sorts of issues but what it can't do and what we've been asking for for a while and still want, is a good explanation on what's suppose to happen if there's actually an emergency key roll?

There's a process somewhere apparently but until we actually know what it is, there's no really way to audit if it's going to work and there's no real way for operators to actually see the process happen and go, "Oh yeah, I understand that is what's suppose to happen and therefore I should I trust the new key." One of the

things people have been talking is just the key rollover but what we also need to be discussing is the algorithm rollover, which is the exact same sort of problem we just went through for this last KSK roll but substantially worse. In the discussion of rolling to a new key, we also need to be discussing rolling to a new algorithm.

RUSS MUNDY: I don't see any comments online, no more at the mic. Thank you, Warren, for mentioning that mail list, I was going to mention that to folks here. It has not been very active lately but I know that it is being monitored closely and when things get posted folks respond, it is certainly being noted by the ICANN Staff, who's working on getting the definition of what's going to be happening next with the KSK rollover? Honestly, I have to say as a member of the Program Committee, I was hoping we would have some input from ICANN at this one but the ICANN Staff that's working on it declined, they said it was too soon to provide anything. Hopefully by our next meeting we will have something of output from the ICANN Staff that's tasked with planning the set of events that are going to happen with the KSK roll next. Keep thinking about what you're looking for, join that mail list if you haven't already and please, post your comments to the mail list and we'll get more reactions there. Thank you,

Tim, again very much for presenting the Geoff set of slides and having a good decision here. Thanks, Tim.

Next up is Wes Hardaker.

WES HARDAKER: In the past, on a regular basis, Viktor Dukhovni and I give a presentation on where we've been with DANE and SMTP's. Before you heard me describing how DANE SMTP works, this is sort of a usage report and last time my author Chip was on left and his was on the right because I did most of those slides. This particular, although a lot of the graphs are mine, Viktor Dukhovni really did all of the routine gathering of data and I'm really just presenting his hard work. I think we're going to have the same scrolling issues as last time; I apologize. Next time I promise to make them high definition aspect ratio, sorry.

I'm going to go over a few things. Little bit of background. This is all about DANE SMTP monitoring and a regular monitoring that Viktor Dukhovni does from Two Sigma. All of this available stats.dnssec-tools.org site. The data is all from Viktor Dukhovni and me putting into a graph and table format is done by some backend crypts that I wrote to publish this on a daily basis. Everyday this page is updated with the latest data that has been polled by Viktor.

Some recent changes. There're new data sources that have been added. Viktor's gotten some more data dumps from a couple of TLDs and other organizations. There's a lot more graphs that have been added, in particular the DNSSEC Growth Graph is now available, which shows actually the number of signed DNSSEC zones. Tables are now sortable as well, if you click on the column name you can sort by the table, it's not very well -- unless you read the text, you don't know that. Since I created this slide there's another update which is that there's not individual graphs of each TLD as well in terms of what's been deployed DANE wise.

In this data set, basically Viktor's code monitors domains delegated from various public suffixes points, notifies operators of issues when he spots them and he gets a lot of information from a various number of sources, the full list is available on the states.dnssec tools webpage. It covers more than 200 million candidate domain names, that number is probably out of date. Basically, he's monitoring DS records, DNS key, MXA, Quad A and TLSA records to determine how well DANE is getting deployed as well as DNSSEC. He captures the certificate change of everything -- of certificates published my MX hosts.

There're 9.87 million domains with DNSSEC validated MX records and that's sort of the same number that was pointed out

in counts counts counts but we're going to augmented it this time with mail specific numbers. There're 1.18 million domains with DANE enabled SMTP and just for reference, six months ago in Barcelona, that number was 300,000, so we've gone in six months from 300,000 to 1.28 million DANE enabled SMTP severs, that is a massive jump. There're millions of users, there's a lot of DANE enabled SMTP servers. There're basically 4,468 zones, which you think is a lot smaller than 1.18 million, we'll come back to that why, there's basically a few providers that are sourcing a whole of SMTP mail servers that are DANE enabled.

A couple of awards. Since Barcelona there's a been a number of interesting changes, one of which is .BANK. Anybody here from .BANK? I was going to give you a round of applause. They went from having a fairly high failure rate to almost near perfect. ONE.COM, anybody here from ONE.COM? Too bad. They signed 707,000 domains since Barcelona and added DANE TLSA records to their zone which is huge. There's two TLDs with basically 100 percent working DNSSEC key validation, in other words, two TLDs with zero errors in them. Anybody here from .BOSTON or .BIBLE or a register associated with them? Finally, large volume TLD .BRAZIL has a very active monitoring system that actually notifies their sub domains and so they have an extremely large number of sub domains that are all signed and properly working at 99.6 success rate. Last call, anybody here from .BRAZIL? Yay,

give him a round of applause. That is an extremely commendable success rate, to get that well of an error rate.

This is the slide from last time, this was what the slide looked like last time in Barcelona. You can see that we're just over 300,000 domains using DANE and SMTP. This next slide is and you can see the white hand -- excuse me. On the right-hand side there's ICANN Barcelona marked with a vertical line and you can see the quantity of jumps that have occurred since there. The biggest one of which is ONE.COM, as I mentioned they signed over the course of a couple of those jumps we're ONE.COM where they got as an SMTP provider, they actually got a large number of their clients using DANE enable SMTP connections with secure TLS. A huge shout out to them for deploying DANE.

If you look at the number of MX hosts, basically the number of providers, these two graphs are actually quite similar, this shows the number of domains protected but, in this case, this is the number of DANE enable providers. ONE.COM is only one tick in this graph even though they've signed 700,000 of their domains or protected 700,000 of their domains. If since Barcelona, again which is marked on the graph, there's still a significant increase coming from near 3,600 to about almost 4,500. In the past six months we've added almost 1,000 DANE protected MS servers, which is fantastic progress.

In terms of number of zones with DS records, in other words these are basically the number of signed zones. I think Viktor's stats are probably the best in the world right now in terms of being able to monitor the number of DNSSEC protected zones. We're coming up on 10 million, that's a pretty impressive jump. This graph unfortunately I don't have the data on the far-left hand side incorporated, so the ICANN Barcelona line is clear on the left but you can see that we've come up yet another million domains signed in the past six months, which is quite good.

A couple quick points about this graph. When you're looking at all of these graphs on the stats.DNSSECTOOLS website, a couple o points. He get's updates on a regular basis from a number of sources and they're not updated daily. That data that he gets can be updated weekly or monthly sometimes. These large jumps that occur because that's when he's getting volume dumps from a couple of sources. Then, there's a slight downfall a lot of the time until the next increase and so the slight downfall occurs because during the course of the coming weeks and months, zones actually go -- they get delated or removed from their registry and so you get the slight falling affect but the overall trend is upward.

A couple of noteworthy points and I mentioned ONE.COM, they deployed 7,600 DANE records. A few other ones,

WEBFORYOU.CZ deployed 27,000. FLEXFILTER.NL deployed 15,000. These are the biggest jumps along with the dates that they correspond to, where they deployed a large number of DANE records all in one approach. If you're an operator of anyone of these sites, thank you very much for your pushing forward with increased email security.

In terms of top DNSSEC TLDs, .NL still is the number one, .COM being second and SE and CZ and VR. The interesting thing about these, if you look at the top 10, they all increased in the number of signed zones that they host, it was only the bottom few that actually changed places. The top 10 are all essentially the same, they all moved up some but none of them actually flipped places. .HU went up, .ORG went up by one and .NU went down by two and we'll come back to those in a minute and .CH went up by one as well.

We're going to dive into some of the new graphs that occur on the STATS.DNSSEC pages, specifically .NL has the highest number of signed domains. Again, this sort of saw tooth pattern occurs because Viktor gets dumps on only a monthly basis. During the course of the month a number of domains start disappearing of start getting removed from the registry and then he gets another bump up and so you can see there's a saw tooth pattern. Be aware when you're glancing at the various pages,

that this saw tooth pattern isn't a mistake, it's just the periodicity of the data coming in.

Consistency is sort of key, this is from .APP, you can see that it doesn't have the saw tooth pattern, that means that stuff isn't getting removed or he's getting daily increments. Same thing is true for .DEV, it has this beautiful nice curve to it, so I put it in just because they are nice and flat and pretty but shows the increasing trend of DNSSEC within the .DEV domain TLD.

A couple of big jumps, .BE had some very large jumps in it where they jumped from 160,000 to 260,000 domains all in one jump, as a provider had actually started signing every zone that they provided for. .DE is sort of a similar case, there was a sudden large jump in .DE over the course of near February of 2019.

Lots of small jumps, .CZ had a large number of domains that just would occasionally get signed, some these had to do with data import consistency as well, clearly some providers were signing new things there.

There's some loss of course too in .NU in particular. A provider that had signed a bunch of stuff decided to remove signing and there's some reasons behind it but apparently, they removed and then re-added at some point, so .NU unfortunately lost a large number of signers as that provider removed DNSSEC

support and then added it again later. That same provider also operates in .SE but I think that in the .SE case, there was more incentive for them to turn it back on, so there was sort of a month where they weren't doing DNSSEC and then reenabled it, so you have this large drop.

A couple of things, just as a reminder on how to do proper DNSSEC so that your graph and numbers look great according to Viktor's monitoring. Keep name server software up to date. Like most software these days, being obsolete is not good due to bugs and issues. Test your zones and specifically look at wild card a or c names are tricky things, especially at apex of the zone. Test zones with empty non terminals and always sign after changing and SOA serial number. When I asked Viktor what his most common seen problem is, that people will go sign their zone and then go update the serial number, which doesn't work because you've got to sign the serial number change. People think, "I'm updating the zone, I need to go change the serial number."

Change the serial number and then sign, apparently that's the most common mistake. Avoid NSEC 3, there's a few zones that turn in it on but that doesn't buy you anything unless you're a really large provider with a whole bunch of zones that aren't signed. It is much better to have NSEC 3 covering your zone or

NSEC covering your zone so that you get that proof of non-existence that I was talking about in my last talk. Avoid NSEC 3 extra iteration counts, it doesn't really buy you anything to use a whole lot NSE3 iterations, so zero is actually recommended count.

Some basic DNSSEC hygiene. Make sure all of your name servers support EDNS 0 and NSEC 3, that includes both name serves and resolvers both. Pretty much most software does these days, so this is sort of obsolete and a couple of years ago it was more important to make sure they provided it. If you're using anything modern, it's mostly likely you already do. Don't block IP fragments, large DNSSEC packets due tend to fragment, so make sure that your fragmentation support is working on your network, regardless whether you're and authoritative server or a resolver.

Make sure you reply with no data or NS domain, not implemented or refused, if you're a resolver in particular. Test for denial existence for each edge case. Denial existence can be somewhat tricky, so make sure that you test your zone handling to make sure that when you are querying for non-existent names you get the answer back that expect, which is that it's not existent and it's proven non-existent by DNSSEC. Then monitor your name server for correct DNSSEC handling. I'll admit, I think

I failed this a couple of times in the past, where I didn't know my zones went out of date. I now very carefully monitor that. There're great plugins for Nagios and other related things that will help you be notified that your zone is about to go out of date.

I think in the future and I would love feedback on this, Viktor and I were talking about putting together a longer presentation for the DNSSEC workshop on how to roll DANE TLSA keys as well as how to do proper algorithm rolling, both within DANE and within DNSSEC. If that sounds like something that would be worthwhile in the further, we'd be happy to put something together. We'll talk to the program committee about it too but I'd love your feedback, if that's a topic you'd like to know how to do.

A couple of important takeaways are that you should always publish your new TLSA records at least well in advance of actually deploying new certificates. One advantage of some types of if you publish your key rather than a certificate, you can actually create a XO79 key even if you don't create the certificate and publish that even before you have the certificate validated by a CA, if you're going to use the CA route, you can publish those well in advance and get the certificate using that key at a later time.

Basically, if you look at the bottom example, in the same way that we often deploy to zone signing key and DNSSEC, you publish one that you're going to switch in the future, you can do the same thing with TLSA records, you publish a future key that you're not yet using, have it ready to go so that when you need to suddenly switch to that next certificate, you can do that just by switching the TLSA server on the MTA and because you've published that key well in advance in the DNS everything will just magically work.

Automate, automate, automate early, automate often is my favorite phrase, even my kids have heard me say that to them. Make sure that you get TLSA records updates and zone resigning, that's all done automatically and key rollovers are automatic as well as acquiring certs and converting to TLSA records. If you're using something like Let's Encrypt where you're rolling certificates on a three-month basis, make sure that you incorporate that into your DANE processing as well. Then, make sure your contacts are working for WHOIS and SOA and Postmaster records so that when something goes wrong, people can reach out and get a hold of you and tell you that they're having issues with your zone or your site.

The DANE resources slide, I'm not going to read all these but Viktor at the last minute gave me a whole bunch of great links

that if you want to read up more on DANE and how to use it and reasons for using 311 style DANE records, as well as some talks that he gave at the New York Lennox Users Group Talk, you might go follow-up on these, you can pull the slides from the ICANN website.

Specifically, Viktor's list of help wanted. More ccTLD lists of signed delegations. Any data feeds that you can give him, ccTLDs that are interested in helping communicate their level of DNSSEC compliance within your ccTLD he would love to talk you or you can talk to me afterwards and I'll give you his contact information. Fixing any DNSSEC issues, make sure you monitor your zones, especially ones centered on denial of existence, he sees most problems in that kind of area. As well as enabling DANE outbound. Even if you don't have a DANE enabled inbound server with a SMTP TLS certificate and deployed DANE, you can still enable it on your outbound sever if you open source software like EXIM or Post Fix, all you have to do is enable it even if you don't have your own hosted domains using it. Of course, if possible, enable it on your own MX servers as well for incoming support as well. Again, thanks very much to ONE.COM who got a whole bunch of stuff deployed in the last six months.

That's it, any questions? Again, if you want to look at pretty graphs, you can look at STATS.DNSSEC-TOOLS.ORG. The very

bottom table is a list of all the TLDS and if you click on each of the TLD names you'll get pretty graphs what each one looks like.

VITTORIO BERTOLA: I think I don't have a question but I have a comment which was part inspired by the long discussion that just went on on the DNS, on incentivizing the adoption of DNSSEC. I mean, I'm happy that we meet at every DNS workshop and we see [inaudible] going up, this is good but, in the end, the basic problem with DNSSEC as we all know is that there are no real incentives to adopt it immediately. Until the moment where it really saves your life, like five years later when you're attacked. There was an interesting comparison made on the list which I found really on point with seatbelts, it's the same mechanism. Seatbelts initially are a cost for a car maker and for the users and you don't get anything back until you have an accident and then they save your life.

But the problem was if you look at the history of this, the only way they could be deploy was first by in few bright car makers pushing them, just saying we want to push this on the market and in the end, it was regulation forced and everyone else had to follow. What I'm thinking is since ICANN is still the place that does both things, brings registries together and then also does

some migration, why don't we ICANN Community ask ICANN Organization to do something for that?

If you have a look at all the TLDs where actually this has been going up, almost all of them have been reaching these targets by giving discounts to people that signed domains, so why doesn't ICANN put a little money to maybe give a discount on the domain fee to registries proportionate to the number of signed domains? At some point and time maybe, ICANN could even require registries to support DNSSEC but still, just using a little money to give some incentives could be useful. Maybe this is something we could think about?

WES HARDAKER: That's a very good question, unfortunately I can't speak to ICANN. I think that the top ccTLDs that you have the most take up are those that actually offered financial incentives to the customers within it and its sort of proven that that model tends to really, really work. The new gTLD program within ICANN mandates that all new gTLDs have DNSSEC signed but they don't sort of force the clients within it and maybe there should be an incentive there. It's a good point. I don't know how to do that going forward. Thank you. Any other comments? I think that brings me to done.

RUSS MUNDY: Thank you very much Wes. We have had a good set of presentations, covering a somewhat wider range of topics then what we have in the past and one of the objectives the Program Committee has had is to certainly be responsive to inputs that we get from the Community. We very much want to hear back from folks that are here, folks that are online and remote and before we do anything further, I'd like to ask if there are any folks in the room that had comments that they'd like to just stand up and give for the Program Community to consider for the next program? What type of things are people thinking about that you'd like to hear presentations and talks and have discussions about next meeting? Any thoughts from anyone?

UNKNOWN SPEAKER: I work for the Global Stakeholder Engagement Team covering the Middle East and one of the questions we keep getting is, what's the intersection between DNSSEC and some of these new technologies such as DoT and DoH and all the other stuff that is still work in progress at the IEFT? Maybe if you can present maybe at the next ICANN meetings, the intersection or the differences between DNSSEC and these different new technologies? Thank you.

WARREN KUMARI: Just to respond to that. Actually, at the last DNSSEC workshop or maybe it was Tech Day, there was a presentation on that, so it might be easy to just point people back at the videos for that. I can't remember what I called it but somewhere there.

UNKNOWN SPEAKER: I'm trying to find it right now but there is also a session, I think it's Wednesday -- the high interest topic, it's put on by SSAC and ccNSO.

RUSS MUNDY: Thanks. We will certainly try to incorporate presentations on new and related material as it comes out. A lot of people engaged in the workshop, in this workshop are also engaged in Tech Day and are active in the IEFT round. This is where we're trying to bring the pieces together and that's a very good suggestion. Other comments we might have from folks as far as high interest, hot topics that you'd like to hear about? Generally, we've kept this focus on DNSSEC but we've also expanded to things that are related to security and the DNS beyond just the straight DNSSEC. This is also something that we'll be including probably in the call for participation for the

next workshop is, what folks are looking at and thinking about in terms of security relative to DNS in a broader sense.

This one of our frequent and regular requests, how folks can help and do. This is something we've included in the workshop for quite a while, it's just a quick summary run through by functional space of things that we think most people that are doing things related to DNS will find themselves in one or more of these roles. TLD operators, make sure your TLD is signed. We've had very good uptake in DNSSEC and the TLDs but there is still some that need to be signed and then not only sign the TLD but also make sure you work with your registrars if you have registrars associated with your TLD, to do DS records, key records, be able to accept what the registrars want to give.

Then stats, the Community is always looking for various statistics to see how things are going. If you are an operator of just a regular zone that's not a TLD, you are likely to be able to do a lot of the same things. One of the differences if you're an operator of the TLD versus the TLD itself, you're going to be likely be working with some type of registrar functionality and this continues to be what I describe as a weak point in the overall DNS picture or DNSSEC usage, where the registrars have had some -- we've had a very mixed up take and registrar support for DNSSEC. Often that has become the long pole in the

tent. When people want to sign their zones, the TLD is signed but the registrar that they're working with does not support DNSSEC. Push on your registrars if they don't support it and ask them for the support. Statistics, always statistics. If you're enterprise, enterprises should be looking at signing their zones also.

There was something that came out of the publications that the US Government does relative to DSN and DNS security and this was included in some of the broader security topics that the US Government is requiring at least contractors that work with the US Government do, that they fall into the enterprise category. They are now required to be DNSSEC signing and doing DNSSEC validations. That's a helpful thing that the US Government did for the broader corporation of DNSSEC. It's something every enterprise ought of look at doing. There may be money and resources involved and so it will take pressure of one sort or another, whether it's regulatory or whether it's needing security or whatever. If you're using an external DNS operator, make sure your DNS operator is offering DNSSEC.

When you're deploying, validating DNS resolver, yes Andrew.

ANDREW McCONACHIE: I have a question from remote from Angela. Is it possible to have a registrar implement DNSSEC while the ccTLD registry does not use DNSSEC?

RUSS MUNDY: Most registrars do support more than a single TLD and in that case there is probably other TLDs that are supporting DNSSEC that are particular registrar works with but if an entity that's in a domain tree, if they're under a TLD is not signed, they are not able to be any more than an island of DNSSEC, they cannot be fully validating if their TLD is not signed. Thanks.

Again, sign your own zones. This is a good thing, even if you're an island, it's better to be signed and be an island than not be signed at all.

Using DNSSEC yourself. One of the things that I used to do at the workshops, don't anymore but I used to actually bring a cellphone that was doing DNSSEC on the cellphone, DNSSEC validation, it would validate the entire chain. You don't have to have a bug huge infrastructure if what you're doing is not supporting a massive number of people. This is one of the reasons why some of us have pushed for many years to get DNSSEC incorporated into applications or at least on the end vice of doing validation on each end machine. There are other

things that does complicate, such as the KSK rollover but it is good to get your validation of your DNSSEC done as close to the end platform as possible.

Work with others, shared what you've learned. Talk at the open forums and privately. This is very helpful. There are several mail lists that have DNSSEC discussions on a regular basis. We want to encourage people to think about doing presentations at our DNSSEC workshops in the future.

Today, before we actually go to the DNSSEC quiz, the DNS quiz I guess it is, I want to especially thank Wes Hardaker and Tim April for the presentations they did for us today. Another round of applause for them. Thank you. Okay Jacques, I think it's over to you now.

JACQUES LATOUR: I finally came back after a lot of requests; we missed the quiz for two ICANN meetings in a row. I hope you're already, I worked on it yesterday. Just quickly, how many of you it's your first time doing this quiz? Did anybody here won the quiz before? We've got one. There's a little bit of myself in that quiz, I don't know why I did that but we'll see how it goes. Behind the program there is a quiz, there is the answers, you need a pen, you write the answer. You have 10 questions on the sheet. I actually have

11 questions, so one of them has two answers, we'll see how it goes. Write your name here and then once we're done the quiz your neighbor can correct your quiz afterwards. Since I don't know much, I can change the rule, I'm right, you're wrong, this is great. It's one point per question, one answer per question. It's a maximum of 10 and a half points, we'll see how that goes.

Ready, everybody's got a pen and a sheet?

Alright, very hard question, first one, what does DNSSEC stand for? A) DNS Security Extensions, B) DNS Security, C) DNS Stock Exchange Control, D) DNS Service Entrance Cable. Easy start here.

Question two, what is the percentage of DNSSEC validation for Africa? A) 7%, B) 12%, C) 18%, D) 21%. These stats come APNIC, you're not allowed to connect and check.

Question three, in RFC7477 a record type specify how a child zone in the DNS can publish a record to indicate to a parental agent that the parental agent may copy and process certain records from the child zone, what is that record type? I'm right, you're wrong. A) Agent Fo, B) The name record, C) C Sync Record, D) DS Record, E) DNS Key record. Whatever I say is right here, is the right answer.

Question four, DNSSEC uses public key cryptography to sign and authenticate DNS resource records that are RSAC, the DNS public key. The public keys are store as a DNS key record, where is that record located? A) In your brother, B) In sister, C) In the child, D) The parent, E) The grandparents. It's a family DNS. Uncle, I missed that, next time, so you got to forget for the next time.

Question five, would a browser enable with DNSSEC DANE TLSA validation be able to detect a website that has been compromised by VGP hijack attack? A) Yes, B) no, C) Maybe not, D) In certain corner cases. My answer is a good answer for that one.

Question five point one, for half a point, this is the half a point question, after that if you got it right or not. Should a browser be able to performed DNSSEC DANE TLSA validation by default? A) Yes, B) No, for a half a point. This is important, you'll see why.

Question six, what does the S stand for in TLSA? A) Service, B) System, C) Systemic

Question seven, which of the following TLD is in the root zone? A) Air BNB, B) .Blockbuster, C) .Cats, D) .Diamonds, E) .Engineers. Which one of the TLD in there is in the root zone? This should be super easy.

Question eight, when do you think Jacques, would like to rollover the KSK again? A) In one year, B) In two years, C) In five years, D) In 10 years. You got to think like me. Whatever is right for that one is me, I'm right, you're all wrong if you don't get it.

Question nine, which of the following DNSSEC related terms in an acronym? A) DANE, B) PKI, C) UDP … ahhh sh*t, never mind. Come on, I got to do better than this. I emailed it to Cathy to work with fixing it last minute -- it's my fault. Anyway, you should know that one.

Last one, which one describes DNSSEC algorithm 13? A) ECDSA128 with child 128, B) ECDSA Curve 256 with child 256, C) 384 and 384, D) 512 and 512 or algorithm 13 assigned.

Alright, now we correct the exam, take your sheet, give it to your neighbor. It's one good point per answer. There is already a couple of freebies in there and one of them is half a point and then maximum is 10 and a half.

DNSSEC stands for DNS Security Extensions, that's good for one point.

21% is the DNSSEC validation for Africa, which is one of the highest on the list, which is good.

Question three, the answer is a C sync record.

Question four, the answer is child, that's where it's located.

Question five, the answer is A, yes.

The next one is, should a browser be enabled to perform DNSSEC DANE TLSA by default, the answer is, yes it should. Eventually we should do must a browser and that will be a different discussion.

5.1, write that after the answer to know if it's right or wrong because that's important for after.

Question six, the answer was already there, if forgot to take it out.

Question seven, which of the following TLDs in the root zone? Blockbuster. I thought they were out of business. But they got the TLD or somebody else doing something? I don't know? How many got that right? Oh, two, three, so those were guesses, I guess.

When do you think Jacques would like to roll the KSK again? 1 year. If you go in the KSK mailing list you'll see an email from me saying we should do it in a year, it's public. I'm right, unless someone is Jacque here. No? I'm the only one, I'm right.

Question nine, initialisms and acronyms are not the same, everybody should know that.

Algorithm 13 is ECDSA Curve 256 Child 256.

Return your sheet back to the original owner, calculate the score, give it back. If your score is an integer, no lunch for you. This is important, the lunch ticket, give it to your neighbor, if said no, that the browser should not default TLSA, no lunch for you. The great DNS guru title goes to --- raise your hand and keep your hand up if you have five or more, that should be easy with three questions that I messed up on. Six or more? Seven or more? Eight or more? There's none. Seven, so we have four winners. Give me ticket. We've got a couple of winners. Four winners. Don't forget your lunch ticket, especially if you go out of the room and come back, you need to have your lunch ticket with you, that's important. Thank you to the lunch sponsors.

RUSS MUNDY: I don't know if it's set up yet. I saw them brining food in and it looks like it's getting very close. After lunch, the Tech Day Program will be in this same room, so more related topics this afternoon. I think Andrew is going out to try to determine if it is close to set up time, we're a little bit early, 15 after I think is what the schedule was. Thanks everybody for coming, we'll look forward to seeing you next time.

[END OF TRANSCRIPTION]