ICANN70 | Virtual Community Forum – DNSSEC and Security Workshop (2 of 3)
Wednesday, March 24, 2021 – 10:30 to 12:00 EST

KATHY SCHNITT: Thank you. Hello, all. And welcome to the DNSSEC and Security Workshop, Part 2 of 3. My name is Kathy and I'm joined by my colleagues, Kimberly Carlson and Andrew McConachie, and we are the remote participation managers for this session.

Please note that this session is being recorded and follows the ICANN expected Standards of Behavior. During the session, questions or comments will only be read aloud if submitted within the Q&A pod. We will read them aloud during the time set by the chair or moderator of this session.

If you would like to ask your question or make your comment verbally, please raise your hand. When called upon, you'll be given permission to unmute your microphone. Kindly unmute your microphone at this time to speak.

This session now includes automated real-time transcription. Please note, this transcription is not official or authoritative. To view the real-time transcription, click on the Closed Caption button in the Zoom toolbar.

And with that, I'm happy to turn the floor over to Mr. Steve Crocker.

STEVE CROCKER: Thank you, Kathy. Welcome, everybody, to Part 2 of the DNSSEC workshop. This section of it will have a panel which has eight speakers, including myself, now. And then Mark Elkins will bring up the rest of it at the end of this hour-and-a-half period.

Our focus in this area is automation of two aspects of DNSSEC provisioning. One is updates of the DNS records, and the other is multi-signer coordination. This panel is a one of a continuing series. This is, I'll call it Episode 4, and our plan is to continue these for as long as it takes until there is sufficient automation. And you'll hear a lot more about all that.

Let's see. I've got to do something. There we go.

So, what I'm referring to here are what I consider to be two gaps in the DNSSEC protocol specs. One is how do you automate the DS update in the event that the DNS provider is not the registrar? And the other is what happens if you have more than one DNS provider and each of them is generating and using its own keys to sign the zone? There has to be some cross-coordination.

Let's see. I see a chat thing that says, "I don't see the slides." Is that the case?

KATHY SCHNITT: No. He's saying on the website. They're on the website. I'm sending him the link. You keep going. You're good.

STEVE CROCKER: Oh, I see. I'm sorry. Thank you. All right.

So, with respect to conveying DS records up to the registry, there are several different ways in which it can be done. The arrows in blue are what I call push approaches in which the new DS record or the equivalent of it is generated at the DNS provider, and then it's pushed up either to the registrant or to the registrar or to the registry. And then, conversely, there are the complimentary pull approaches.

The only one of these that is extant that we have in the field actively being provided is the pull approach where the registry looks into the DNS zone and sees CDS or CDNSKEY records and pulls those up. But the others are theoretically possible. And, in particular, one of the possibilities is to have the registrar do the pulling instead of the registry. And in the next talk, we're going to hear an update from Brian Dickson about the approach at GoDaddy.

With respect to cross-signing when there are multiple DNS providers, there are, again, more than one approach that are possible. But one of the approaches that does not seem to be likely to happen is where the DNS providers directly interact with each other. Instead, it looks like it will take some degree of coordination from an external piece of software or service to do that. And we will hear a great deal more about all of that. Though, as I say, that's the most likely.

So, today's agenda. We're going to have Brian Dickson momentarily, and then we're going to have a talk on the multi-signer project foundations. We have some very exciting work underway involving several people that you're going to hear from, implementations.

And then Eric Osterweil is going to talk about two aspects of studying these transitions that have to take place. One is, how do you observe whether they are happening? And the other is doing an analysis of what kinds of transitions work and what kinds don't.

Along the way, a couple of tangential issues have come up. One is, how do you provide guidance to the implementers on when to transition from one algorithm to another? This is a broader issue that extends beyond these two topics and is appropriate for DNSSEC as a whole.

There is interesting work and important work going on in DNSOP in the IETF in other areas. I mention this here only to suggest that, in future panels, we'll try to say a bit more about that.

The other thing that has come up is that in studying the existing RFCs, there is an indication that the same algorithm is supposed to be used in some areas where we think it is appropriate for that to be looser and for there to be an opportunity to have different algorithms in place for one DNS operator to have one algorithm and other to have the other. And so, we will be spending a little bit of time not during this session, but over the next months trying to examine the older RFCs and see if some update of those is necessary.

So with that, let me turn things over to Brian. I'll run your slides for you. I hope you're there. Please take it away.

BRIAN DICKSON: Good morning, good afternoon, good evening depending on where you are. This is a very brief update. This diagram gives an indication of where the pulling element from the three scenarios resides within the possible set of models for doing updates. I think this a very good diagram for reference, but we should probably go to the next slide. Yes.

So, this is all relevant for the KSK rollover. The first two scenarios are when GoDaddy is doing the managed DNS, and the third one which is the proposed enhancement is where DNS is being provided by a third party and GoDaddy is the registrar. And the requirement is that there needs to be a method of pushing the updated DS to the registry, and the method for that is to pull for CDS and CDNSKEY in the child zone, and then send the EPP update to the registry. The benefit there is that it does not require any changes on the registry side.

So, the update is that while this is part of GoDaddy's evolving DNS strategy and we have not made any public commitments on product, including what it will end up being part of or if it will be standalone or any specific date or any pricing, the update is that it is now under active development from the technology front.

STEVE CROCKER: That's tantalizing. Let's see. I think there's one more slide here.

BRIAN DICKSON: Yes. And this just gives a good update here. So, under development is now very much a going concern. I wanted to make that clear.

STEVE CROCKER: Let me press you ever so slightly. So, you said you can't make a commitment to actual dates of operation. Is there anything that you can say? We're going to inescapably invite you to speak again at the next panel which is in a few months. What are you likely to be able to say by then?

BRIAN DICKSON: At that point, I'm hoping to be able to give an indication of status of development itself, and potentially ask for volunteers to participate in our OTE, Operational Test Environment, for being able to actually test this out. But, again, that's really just tentative at this point.

STEVE CROCKER: Oh, I can hardly wait.

BRIAN DICKSON: Good.

STEVE CROCKER: Thank you very much. Much appreciated. When you do this, of course because it's GoDaddy, I expect that it will have a very substantial impact across the whole environment. So, this will be important. Thank you.

BRIAN DICKSON: Thank you.

STEVE CROCKER: So with that, let me move on to the multi-signer project. Shumon and I have been looking at and talking about and encouraging a lot of work on this area. Shumon's been doing real work. I've been doing a lot of jabbering. So, the floor is entirely yours, Shumon.

SHUMON HUQUE: Great. Thank you. You can hear me okay?

STEVE CROCKER: Yes.

SHUMON HUQUE: Great. So, thank you, Steve. I'm going to briefly summarize the multi-signer project that a team of us on this panel are engaged in. Next slide, please.

So, at a high level, what we want to do is to tackle the general problem of the operation of a DNS zone cooperatively across multiple distinct DNS providers or operators where each DNS operator has their own set of signing keys. We want to also include the topic of DS automation in this type of configuration. So, one of the ideas is that if we tackle this more general and slightly more complex case of DNSSEC configuration, then other configurations and processes will naturally fall out.

So, we'll certainly consider the case of steady state operation of a multi-signer configuration, but also transient states which are very useful for enabling signed DNS zone handoff from one operator to another in a manner that is entirely non-disruptive. That's to say we should be able to do this such that at no point in time was the DNS zone insecure or unresolvable, and that the DNSSEC chain of trust was unbroke, fully intact, during the entire process [of] migration.

So, this one is a little bit of a problem in the industry today, as some of you in attendance may already know, where a frequent mode is to go insecure for these kinds of transfers. So, it would be nice to have a well-understood and tested procedure for doing this. Next slide, please.

So, we're currently engaged in developing some prototype testbeds running multi-signer configurations. So, at the last ICANN DNSSEC Workshop, I spoke a little bit about some small-scale prototypes that I had developed myself, but what we really wanted to get to was larger testbeds involving actually multiple distinct parties. And that's the stage we have now evolved into. And a couple of upcoming presentations by my co-panelists will go into more details on those.

So, one of our goals in the course of operation and setup of these testbeds is to develop documentation about configuration, operational processes, and other recommendations. We'll be investigating multi-signer support and/or deficiencies in open-source software and also in commercial managed DNS providers to see what works well and where we can do better. And where we might need to do better, try to work with the involved parties to make that happen. We want to investigate and/or develop multi-signer toolkits that help a zone owner to deploy and manage a multi-signer configuration.

And then, of course, as Steve alluded to protocol issues, we want to identify any gaps that might exist in the DNS protocol specifications in this area. And, if necessary, propose enhancements to close any identified gaps.

And I seem to have neglected to put it on the slide, but another important aspect of this project I did want to mention is measurement. So, DNSSEC involves, necessarily, key transitions and subtle timing aspects that are useful to measure not just in any kind of targeted testbed that we set up, but also ecosystem-wide measurements, too. And Eric, as you heard, later in this panel will talk about that topic in some detail.

So, I think that was it for the general introduction for the project. So, Steve, shall we just move on directly to the next presentation?

STEVE CROCKER: Yes, but I see a question that I want to respond to because I neglected to say something as I opened.

The question is, “Are there tools currently available to determine whether a TLD operator is automating the publication of a DS to the parent zone?” Dan York, in his opening talk, showed an enhancement to the existing DNSSEC deployment maps that are coming out that are now going to show which ccTLDs on the map and in the database—it also will cover gTLDs—which of them have a mechanism for updating the DS record automatically, almost uniformly, using CDS and CDNSKEY polling. But, the idea of the map is more general.

So, that's at least a partial answer to that question. Thank you.

So, I'll go on to the to the next slide. Now this is, Shumon, you and our valued colleague Ulrich Wisser from the Swedish Internet foundation. And there he is. I assume Ulrich's going to be the speaker.

SHUMON HUQUE: Yes. I mean, I think we're just going to do it jointly. So, I'm going to do the first half, and Ulrich will jump in and do the second half.

Okay. All right. So, next up we're going to talk about the multi-signer protocols a bit. Next slide, please.

We'll cover two things mainly. First, an overview of the existing multi-signer specification. Now we've done this before at past workshops, so for those who've already seen some of this, this will be a little bit of a refresher. I think that was in Episode 2, to use Steve's characterization of this panel series.

But specifically, what we'll focus on this time is Model 2 of the specification. That's where each provider has their own set of signing keys—KSK, ZKS, etc. And we won't talk about Model 1 where the zone owner controlled a shared KSK. And the reason is that Model 2 now appears to be more generally useful and, in our view, more likely to be implemented and adopted. Also, it supports combined signings keys or CSKs. And we have a participant organization in our testbed that does in fact use CSKs.

And lastly, this model is directly applicable to the topic of DNS operator handoff, as we've already mentioned.

The second topic is about a new IETF draft specification that we've introduced about automating multi-signer configurations. And Ulrich will talk about that in a bit. Next slide, please.

As you may be aware, the spec is described in RFC 8901. And the rest of the items on the slide, I've already touched on. So, why don't we just move ahead to the next one.

And I'm going to now walk through a series of diagrams for Model 2 of the multi-signer DNSSEC configuration. So, here we have a situation where each provider has their own set of KSKs and ZSKs. So, there's Provider A on the left in blue and Provider B on the right.

The zone owner uses provider-specific APIs to update zone content identically at each provider. And, importantly, the zone owner has to coordinate the cross-sharing of the zone signing keys between the providers. Next slide.

So, that means that ZSK from Provider A needs to be imported into Provider B. And just to clarify, this is the public portion only ZSK. The private key corresponding to ZSK remains guarded at the respective provider. Next slide.

And vice versa. Provider B's zone signing key goes to Provider A.

So, at this moment this requires the zone owner to manage the cross-sharing using key export and import APIs offered by the providers. The second draft we're developing will look into automation protocols to make this less manual. Next slide.

Then, finally, after the cross-sharing is done, the DS record in the parent zone needs to be updated. So, specifically that means that the DS RRset in the parent has to reference each provider's KSK. So, there needs to be installed a DS record for Provider A's KSK, and another one for Provider B.

So, once this configuration bootstrapping is in place, DNSSEC will just work. There are some additional operational tasks when you're doing key rollovers which are a bit more involved with multi-signer, but nothing really earth shattering. I'm going to omit those details in the interest of time, but I will mention that part of the work we're doing in the testbeds will flesh out the detailed procedures for some of this. Next slide.

So, I think … Let's skip on to the ... Yeah. I think there was one more slide. Wasn't there?

STEVE CROCKER: I don't see it.

SHUMON HUQUE: Right after the end of the diagrams. Okay.

So, what I wanted to say before handing it off to Ulrich is that RFC 8901 describes generally how multi-signer protocol works, but arguably omitted a few things, or at least kind of hand-waved over them. So, it does not, for example, describe detailed operational procedures for initial bootstrapping, adding and detaching providers, or mechanisms for automating the entire configuration. So, that's the topic of the follow-on draft.

And with that, let me turn things over to Ulrich.

ULRICH WISSER: Hey. Thank you, Shumon. Thank you, Steve. Next slide, please.

CSYNC, yeah. Yeah. So, we'll be looking at the protocol specifications here. We'll be looking at … And we will have two providers running the zone independently. Each of them has their own set of keys, either KSK, ZSK, or the Combined Signing Key. The data synchronization is out of scope for this document. We believe that the zone owner will do that in the background with some other mechanism. Next slide, please.

So, what will happen? I think the RFC for the CDS/CDNSKEY describes how you actually can set up an initial trust for one provider. And that is what we expect you to do to start a multi-signer setup. So, you will have one provider who has already a DS record in the parent and is signing the zone with the respective keys. And now we want a second provider to join the first one in this multi-signer setup.

So, what, then, these providers have to do, as Shumon pointed out, they have to exchange the CSK. And we have to wait for some details to distribute that on the Internet. Then we can actually compute CDS/CDNSKEYs. And here you see that the providers cannot do this on their own because they don't have the complete KSK set. So, this has to be done by the zone operator or zone owner, you could say.

And then we publish CDS/CDNSKEY records at both providers. And then, of course, we have to wait again. And especially we have to wait for the parent to pick up the CDS/CDNSKEY sets and update the parent DS set. And once the parent is updated, we can remove the CDS/CDNSKEYs. Next slide, please. Oh, that was one back, please. No? Yes.

So, now we have an … Actually, the parent NS records still point to the first provider, but we have a key set and DS record that would actually allow us to use both providers. And now we need to update the NS set.

So, first of all, both providers need to exchange the DNS records. And then we can put in the CSYNC record that has the NS and the A and AAAA set which allows the parent then to update the NS set at the parent. And if needed, import any glue. And then we again wait for the parent to update, and we can remove the CSYNC record. And then we have a complete set up of a multi-signer.

And if we wanted to have, let's say, three providers in this multi-signer setup, we would have to do this again for a third signer. But we have to go through this algorithm for every signer joining in a multi-signer setup. Okay? Next slide, please.

Okay. There were actually some slides saying how we leave this. But, okay. It’s okay. I can just [power through this]. It’s okay.

So, in the draft you actually have a description of the algorithm how you actually then leave if one of the providers wanted to leave this multi-signer setup. Because then, of course, you cannot just switch off your name servers. You first have to remove your name servers from the setup, and then you have to remove your keys in a controlled way so as to always keep the validation and the trust chain intact.

And then we have this … Yeah. We have a few things that we still have to do. We have to make a specification for key rollovers, especially ZSK and KSK rollovers. And the most more interesting case is the algorithm rollover specification because it has …

There isn't a case where you do it very similar to how you do it in a single provider setup, but then there is a case where you want to change operators. And these two operators might not support the same algorithm. And that is something that is not addressed in the standard DNSSEC RFCs. And so, that isn't one of the points where we have to do a lot more work to see how this is going to work.

Maybe some of you are on the DNSOP mailing list and have seen my inputs on the lax validation which has actually to do with [inaudible] [at this point]. And maybe Shumon will want to say something about this point, too.

SHUMON HUQUE: Yeah. So, I think there's another slide which specifically talks about the multiple algorithms, I think, after this. Right? Or am I mistaken?

ULRICH WISSER: I believe this is the old slide deck, actually.

SHUMON HUQUE: Oh, I see. Okay. This is the last slide?

ULRICH WISSER: Yes.

SHUMON HUQUE: Okay. So, I don’t really have any specific things to add other than to say that … Yeah, so this is a problem for both the multi-signer configuration. You can't, at the moment, run a multi-signer configuration where each side is using a distinct algorithm. Right? That doesn't work.

And then naturally, as a corollary to, that means, of course, that you can't non-disruptively transfer a signed zone from one operator to another without tweaking the DNSSEC protocol specification. So, as you said, you've commented on relaxing the validation rules a little bit in the specification to allow things like that to happen.

So, the other thing I would mention is that introducing some signaling to relax or not may be useful because there may be other cases where a provider using multiple algorithms may want downgrade protection for whatever reason. I think that topic came up, too, in one DNSOP discussion. But really, from the point of view of a DNS zone operator, only they know their intent with using multiple algorithms. Right?

So, there should be a way to signal that intent, which there isn't, in the DNS protocol today. Right? Maybe they don't care about downgrade protection because, specifically, they want to deploy a multi-signer configuration or they want to transfer their zone from one operator to another operator where the operators are using disjoint algorithm set. Right? So, really, I think some signaling is missing, as well as the revisions to make the validation more lax.

STEVE CROCKER: Does that do it?

ULRICH WISSER: Yes, thank you. I think we are …

STEVE CROCKER: Thank you very much. It's a somewhat complicated business, and I hope that it has all come across. And I apologize if I put the wrong deck in. I don't know where the where the miscue was, Ulrich. We'll sort that out next time, perhaps.

All right. We'll move on to Peter Thomassen from Secure Systems Engineering who has joined the group. We've all been working together trying to develop testbeds and do the experimentation. And so, Peter, I'll turn it over to you.

PETER THOMASSEN: Yeah. Thank you, Steve. You have a question in the Q&A for you. If you want to take that later, that's fine. Or now.

STEVE CROCKER: No. It's too complicated.

PETER THOMASSEN: Okay. All right. So then, while I talk you can think about an answer. Okay. So, I'm Peter Thomassen. I'm the founder of the deSEC, and we joined the multi-signer effort and are part of the testbed. And since the project, probably—I mean the deSEC project—is probably unknown to most people here, Steve suggested I introduced the project and say a few words on how to use our API to facilitate multi-signer. Yeah. Next slide, please.

So, very quickly. What is deSEC? We are a free DNS hosting service designed with security in mind. Our mission, essentially, is the same as the mission of Let's Encrypt except for DNSSEC. So, we're public and free to use. And anybody who has had trouble to use TLS certificates in the past is going to Let's Encrypt. And if people's DNS providers don't support DNSSEC or only in a very restricted fashion, they can come to us.

We have all automated DNSSEC and a nice API and graphical user interface to make it easy for people. We support modern stuff like DANE records, TLSA, or long OPENPGPKEY records for a PGPKEY exchange. And also, there is new stuff like HTTPS and SVCB records with which you can do things like aliasing on the apex which you can't do with CNAMEs. We also run a dynamic DNS service under dedyn.io which also has DNSSEC.

We have been developing for quite some years because it has been a side project, initially. And we launched last year in April, and since then started hosting a few thousand zones. We got some inquiries from top-level domains, but that is still ongoing. And we have become an active community member, for example, with this multi-signer effort and also with other standardization efforts such as the Catalog Zone specification that had a talk two weeks ago at the IETF 110 meeting in case anyone saw that.

And we are supported by Secure Systems Engineering, a Berlin-based IT security consultancy that supports us with the cost for most of the infrastructure. I have to say that I work at this company, and it's great that my employer supports us this way. But, still, we're looking for other partners as well, especially additional AnyCast resources or developmental partners. Next slide, please.

Yeah. So, why are we here? I think it's very quick to say [that] because of the state of DNSSEC regarding adoption and barriers is not too good. The appeal is great, though, and we just figured it's time to have a DNSSEC provider out there that's really free and open-sourced, feature complete and stable. So, I think we can rely on it.

On the next slide, there is a quick overview of how it works. I'll not going to any details, just very roughly. Our DNS engine is PowerDNS, so we support pretty much anything that PowerDNS supports, especially regarding record types. Our API application is a Django application. And then there's all sorts of internal stuff going on that I'm not going into.

deSEC is a nonprofit organization and registered in Germany, so we made sure all the keys also stay in Germany—we don't have them [on our frontend PoPs]—and are subject to GDPR and all these data privacy laws from the European Union.

We have two AnyCast networks. One has seven, the other has eight points of presence and they're capable of serving roughly 1 million zones, given the current load of the ones that we have.

I'll spare you the replication details. And the user interface is done in single page applications [done] with Vue.js centralized and the REST API which, in turn, has advanced features like paging and atomic transactions for several records at once. You may want that for multi-signer, too, if you provide, let's say, CDS and CDNSKEY at the same time without inconsistencies.

On the next slides we have a very quick overview of the AnyCast map. So, this is distributed across ns1 and ns2 which are under different top-level domains. And we're present in most regions of the world. Yeah. Next slide, please.

This is a very quick overview of how you add a record. So, this is an MX record in this case. Of course, it would be very similar for the multi-signer effort where you have to add extra DNSKEY records, so we make no difference in adding multi-signer DNSKEY or CDS/CDNSKEY records, as opposed to any other record. We just merge them with the ones that we automatically provision for our own signing.

You can see the web interface here. This is the mobile view, and if you enter things with multiple fields such as MX records, you get field-specific validation. But at the same time, the input field is also capable of accepting the whole MX record string so you don't have to do the division yourself into the different [fields]. You can paste or type through or do it individually as you prefer.

The REST API is documented at desec.readthedocs.io. We have a very quick demo of that because this is how the multi-signer testbed also uses the API for provisioning DNSKEY records.

So, Steve, if you would like to proceed to the next slide, there is demo time. There's going to be a one- or two-minute demo, and then we're on to the actual multi-signer testbed reports.

STEVE CROCKER: I think this is it.

PETER THOMASSEN: Very cool, thanks. So, we're using a tool here that's called HTTP. It's like cURL except the syntax is easier. We're doing a POST request or API. Some token authorization. And here comes the post payload. And we create a domain called cloudapp.example. Here you go. DNSKEY and everything is here. And you can query it immediately from our name service, including the signatures.

Now we create records. We'll create two records at once, so it's a "bike" request. And what we do is send a JSON list of two record objects. The first is of type HTTPS because that's new. And so, we do an apex alias and the record is going to point us to gateway.cloudapp.example which is the sub-domain that we are creating now in the same request. We'll add a gateway sub-domain of type A for ipv4 address. And we'll add two domains here, sorry, two IP addresses here.

Now you can see that there were some validation errors. So, for the first one, the sub-name was missing. And for the second one, the type was incorrect. I typed "AA" instead of just one A." So, you get validation feedback specifically for the things that are wrong. And the GUI also use that to assign things to the display.

So, now we correct it. We still get another error because you forgot the final dot. And now that we have that, the two records are provisioned. We can immediately query them. This is the other AnyCast network using dig. Dig doesn't know HTTPS yet, and if we do that we get the record back including the additional section with two A records. Yep. So, that is the deSEC API. This is used, then, by Ulrich in the deSEC testbed.

And I will quickly show you one more slide on what we actually did at deSEC to facilitate multi-signer setups. So as was mentioned earlier—exactly, this slide.

As was mentioned earlier by Shumon, it's necessary for multi-signer to publish the other party's public keys, or the [hashes] of that, the CDS records. And we automatically publish our own ones already. So, we have DNSKEY and CDS/CDNSKEY routinely for all zones, and we use the CSK combined model.

Now we also support additional provisioning of extra DNSKEY records. You can add those just like any other record. You don't have to worry about the automatic ones. You just provision the manual ones and then when there is query time, the response will be generated by merging the automatic and the manually provisioned ones. Yeah.

For fully automatic migration of names of providers, you have to signal to the registry that the NS RRsets changed. And for that, CSYNC is needed. And that is not yet supported because PowerDNS doesn't have it yet. However, I think it's going to be in the next release. And it's really cool that PowerDNS, I think, is so far advanced with supporting all of this stuff, for example, like adding additional DNSKEY stuff. That was really easy for us to expose to our users.

So, this is all to say about deSEC. And Ulrich, I believe, is now going to talk about some practical results from the testbeds with deSEC and other providers, too.

STEVE CROCKER: Thank you. Let’s see. Getting back here. Okay. Back to you, Ulrich.

ULRICH WISSER: Thank you, Peter. Thank you, Steve. Next slide, please.

So, what we actually tried to do here is that after … In the last presentation you saw me going through the algorithm. We thought it would be cool to actually do that in the real world—set up some domains and go through the algorithm, see how it works.

And then, obviously, you need a parent or a registrar. Somebody who does the CDS/CDNSKEY scanning. And you need somebody who does the CSYNC scanning, too. And I actually don't know anybody who does that right now. That was actually a question I wanted to ask Brian—if they have plans to support CSYNC in the future. But I couldn't put that in the box, so we can come back to that.

Okay. So, what we did is we did actually register domains in the .ch and the .cz zone because they support CDNS scanning. And we also registered in the .se zone, but we will get the scanning probabilities sometime after Easter. Okay. Please, Steve, next slide. Yeah.

So, so there's actually two modes. You can do this. And the one that is easier to implement is a centralized one. That is, you have a zone owner or controller who does the synchronization between the different signers. Okay. And so, the controller takes care of the data exchange of the keys, the CDS/CDNSKEY records, and takes care of the timing for the publishing and removal of records. Next slide, please.

And so, we had initial success, actually, with deSEC.io because the API allowed us to put in the CDS/CDNSKEY records very easily. Unfortunately, we had a really hard time doing this with any name server software [not by] PowerDNS. And that is basically because … What we tried is … We tried Dynamic Updates, and through a Dynamic Update introduced the DNSKEY record in the zone. And that was just denied by any software. That’s a no-go. And the same is true for CDS and CDNSKEY records.

And then we tried to do this with the command line tools that the name servers provide, and we actually managed to do that for PowerDNS. And we got some help from ISC for BIND, and from the check registry for Knot, but we have not fully succeeded there. And one of the problems is that the name servers calculate their own CDS/CDNSKEY set and they will not allow you to put in any other additional records.

So, we're working on this front and I think we have to do some communication with the producers of the software. Yes. So, next slide, please.

And then we have the distributed version. You can do this. And that is some version that we envision in the future where, actually, these name server softwares, instead of having a centralized controller, talk to each other and exchange the needed information and timing information, and then do all the whole process by themselves so that no centralized actor is needed. And that is, of course, something that the other registries for .se are very interested in because that would allow for not so technically savvy domain owners to change names of operators without getting involved. Next slide, please.

I think that was my last slide. Yes. So, thank you. Oh, there was a question in the last, I think. Wes asked why we would remove the CSYNC record. And, no, you're totally right, Wes. You could leave the CSYNC record for all time and just change the NS records when needed. So, yeah. There's not really a need to remove the CSYNC record. It's just a convenience.

STEVE CROCKER: Thank you very much. So, we'll move on to Eric Osterweil George Mason University for a pair of talks, the first on DNSKEY Transition Observatory.

ERIC OSTERWEIL: Yeah. Hey, everyone. I know some of you. I don't know others. But, yeah, I'm going to talk about things and I'm going to try and keep my speech slow. But I want to also make it punchy. So, apologies if I overdrive the channel. So, click.

Shumon gave me a great shout-out about doing some measurements, and so these are going to be very measurement-focused pieces of work. And this one's talking about a project that's coming up that we're in the process of doing with Priya Ravichander, one of the students at George Mason, and Steve and myself.

And what we're focused on, as with a number of things in our lab, is when keys change end zones. And so, we call this a transition for reasons that I think I'll talk a little bit more about in the next presentation. But, in short, when a lot of people talk about rollovers, we sort of [feel more general cases is a key transition]. And we feel like this is kind of a developing field of study, so we're actually pretty excited about it.

In work that we've been doing, we've focused a lot on specifications as they meet measurements that we've taken, and we're real excited about working with the Shumon, Steve, and everyone on the call with the multi-signer work because it's like the next stage of things that are coming.

But, regardless, back to this, the DNSSEC key Transition Observatory. What we want to know is, in addition to what people are doing and choreographing and expecting, what a resolvers actually seeing? And so, that's underlying what we're doing with this. Click.

So, in short, I think everyone probably understands this perfectly clear, but being an academic, I'm kind of pedantic sometimes. So, the simplest case of a key rollover or key transition would be of key A. And then you add key B, but you still sign with key A. And then at a later time, you sign the key set with key B. And after that, you remove key A. So, key transition is simple in theory. Click.

But in theory, there's no difference between theory and practice. But in practice, there is. So, here are some measurements that we've taken for the last 15 years of key transitions that we've seen. These are KSKs, and you can see they go back to when we first started monitoring and 2005, actually. But the transitions we see didn't start until 2006 and all the way up until last year. You can see there's a lot of varying approaches that people are taking. We just basically said, "Based on our measurements, we can actually detect which key transition procedure you're following from the RFCs." Click.

And here are the ZSKs. And what's, I think in my opinion, a little more interesting about this one—and keep an eye on the time, but I see that I'm punching along rather quickly—is that we see a lot of initial adherence to standards. And then lately, there are these unknown types of key transitions. And what that just means is that what's being done operationally is not conforming with what the RFC has prescribe probably because of software optimizations. Click.

But back to what we're doing here. So, in the observatory our plan is to basically say. “Yeah, well the long and short of managing and analyzing what a name server is doing is great. It's important. But what are the resolvers are seeing? Are the resolvers getting the hint?” And this is a question that really is a distributed proposition. My resolve might be working, and if yours [is and that is], then great. For you. It may not even be noticeable to me.

So, there's a lot of things that we have to factor into this measurement operation. We have to factor in timing. We have to factor in caching. And now with multi-signers, we have to factor in, are there are multi-signers involved? And so, what we really want to do is, rather than prescribe what's going to happen, we want to take a look and build an observatory so we can actually do what we did before with SecSpider. We can actually take some conscientious measurements and then be able to take a step back and say, “What was actually going on?” in addition to, “What did we think was going to happen?” So, that is the motivation for building this distributed observatory. Click.

And what we imagine is something like this. So, this is all this is vaporware. I’ll tell you. Completely, truth in advertising. But this is what our intention is: to use a distributed monitoring platform to basically go off and say, for example, our resolvers, what are they seeing around the world?

So, for example, if you think of the picture I had earlier. You start off in the theory simple key transition with key A. So, then you would hope everyone around the world that's doing DNSSEC resolution is seeing key A in. Click.

And similarly, when you get to the next phase, when you introduce key B but you're not using it, you still really hope that everyone is seeing key A as being K is being in use. Click.

But then when you when you transition over and start signing with key B, you wouldn't expect, necessarily, everyone around the world to see that all at once. So, you probably see some heterogeneity in periods during the transition. Click.

And then eventually what you'd like to hope is that when you have finished your transition and you're now working just with the new key—the remaining key, key B—that that is what everyone's seeing. Click.

And our observatory will basically start to look at how to answer the question, "How long until this is a problem?" No offense to anybody in the region that is falling behind in my picture. This was a random choice.

But, yeah, so this is just sort of the tool that we want to build to be able to do this analysis and answer these questions and take the lessons we learned forward. Click.

So, our intention is to build this as a tool that operates in real time. In other words, we want it to be able to be useful to people that are operating real infrastructure so you can go and track, for example, a zone that you operate in this observatory and see how it works in real time.

And what we want to do is we want to basically, like I said, codify what is acceptance criteria that we can maybe even push forward into operational or standards forums. In other words, when we see that one lagging region, or [*n*] lagging regions, and for how long they're lagging, can we say something quantitative like, "This indicates a problem, or this just indicates how long it takes to do stuff, or how long does a transition actually take"?

And, of course, we'll compare that against things like TTLs, etc., that are codified in the records. Click.

STEVE CROCKER: Is that it, or next slide?

ERIC OSTERWEIL: Yeah, next slide. So, I think I went way under budget on time, but that's good. I'm good at getting us back on track. Hopefully, that wasn't completely unclear. But this is something that we're really excited about doing because I think it works really well with some of the earlier presentations where we want to keep an eye on things as they go far.

STEVE CROCKER: Thank you.

ERIC OSTERWEIL: Okay. So, me again. So, this is some related but separate work that I've been doing with a separate set of collaborators listed here: Pouyan, Thomas, and Matthias. And I think some are on the call.

But this is basically more substrate for what I was just talking about. When we actually talk about these key transitions, what am I talking about? Click.

So again, this is all measurements-based. So, we've been watching DNSSEC since pretty much the beginning of its current standards in 2005. And when we started doing this, we kept a lot of data around it. And we certainly didn't have any idea that, for example, this is what we would be tracking years down the road.

And recently, with the root KSK rolling over there, has been sort of a surge of interest in tracking key rollovers. But one of the things that we observed from looking at the data is that the word "rollover" maybe isn't the exact right word to use, at least in our opinion. And so, to summarize this presentation, I'll steal the Chinese proverb, "To know the road ahead, ask those coming back."

In other words, we've actually seen key transitions in DNSSEC in the wild for 15 years. So, what can we learn by looking at those, and is there something we can formalize out of that? Click.

So, to my main point, the key transition. Our central observation is that changing from one key to another is not really the only way that DNSSEC zones transition their keys. So, for example, if I have a zone that has more than one key in it before it starts its transition or has more than one key in it afterwards, then if I'm transitioning from, instead of one key let's say *n* keys to, let's say, *m* keys. Then, which ones did I just transition to? Did all the keys that remain get transitioned to? And did the key that left, if there was more than one key that left, did they all transition to all the remaining keys?

Once you stop talking about a one-to-one transition—a departing key and a remaining key—it's really hard to know who transitioned to whom and which keys were in use at what times. So, I've kind of buried the lead with these three bullet points, but these really underscore why we think "transition" is the right word better than "rollover."

And this histogram on the side, or this scatter plot on the side basically shows that there are a lot of zones that we've seen over the years that have multiple keys involved in this transition process, and it's really hard to say this one key went to this one other key.

So, that's why we basically, one, tried to general the term a little bit. And that's why we talk about transitions. But two, we also decided that it's really important to know what should you measure in order to actually understand what's happening with a key transition. Click.

So, we've basically come up with a proposed anatomy for what it is that is important to look at, to measure, to understand about when keys are transitioned. We call it an anatomy because we've broken the pieces down and given them a bunch of names.

So, one of the things that we keep track of is the relative timing of when DS records show up with respect to the key that they're talking about if the key has DS records. Then there's the various overlap periods. There are periods where keys are pre-published. There are periods where keys are both signing. Of course, the keys being used is an important part of the key transition. If a key shows up and it's not being used, that's different than if a key shows up and it's being used to validate data.

So, we've broken this out into several measurable pieces. We also include in our anatomy how many keys are involved in the transition and what the relative ages are. For example, when a key shows up and then gets yanked out again and there's an older key that remains, that's different than when a newer key shows up and succeeds and older key.

And so, this is great but what we did with that is we went and we looked at RFC guidance. And we basically use this anatomy to be able to detect which type of guidance from which RFC was being followed at any given time. And what we're really happy about was that the anatomy was general enough that we could also go to the academic literature and we could apply the anatomy to codify that. And we're looking forward to doing the same with RFC 8901. Click.

So, I sort of tease these histograms on the last presentation, but here they are again just more in context, perhaps, which is where we actually applied this transition anatomy and actually measured how frequently people were using different types of RFC guidance.

And again, this is a view of DNSSEC’s life, going back to the beginning of its recent round of standardization all the way up until recently. We still do ongoing measurement, but these were cut off at 2002 because most of the world stopped in 2000—or 2020. Click. That was a joke. Okay, anyway.

So, I think I’ve probably blown my time budget, but it's really hard to stop talking. So anyway, I think our observation is that everyone's getting much better about DNSSEC and more comfortable, and the tools are getting more mature, which is great.

And so now, what we can actually do is look at these last 15 years and look at trending and what has become popular in the way that we've gotten used to managing our zones, and we think our anatomy is a really useful tool for doing that. And our hope is that by looking at what people have been doing and are continuing to do now, that this might be useful in providing guidance and in bringing ourselves into the guidance process at the IETF and here and other places.

This is the product of the paper that's under submission, but I’m happy to share it if anyone wants to ping me [inaudible]. Click.

STEVE CROCKER: Thank you, thank you, thank you. Everyone stayed well within their time, and so we have time for questions. I’ve tried to answer a couple along the way, but we have a several minutes here for open freeform discussion/commentary, whatever we want, or for additional comments from any of the panelists or presenters before we move on to Mark Elkins. I’ll just sit quietly and wait and see what happens here for a few minutes.

ULRICH WISSER: If nothing else comes up, I'll jump shortly in here and would like to say thank you to Peter Van Dyke because when we requested the support for CSYNC, he was very fast to provide that. And we have now a [built] version of PowerDNS that supports CSYNC in our testbed. So, that was well worked and we thank you for your support.

STEVE CROCKER: Good. There was a question in the chat. "Do we know what the current DNSSEC adoption rate is by ICANN region, and how that growth rate is expected to behave like?" Dan York showed DNSSEC adoption rate overall in his opening talk. I don't have any statistics. He might have them broken down by region. So, that's an interesting question that perhaps he or others can address.

ERIC OSTERWEIL: As one data point, I wasn't able to attend Dan's talk, so I'm not sure if this is repetitive with what he covered. But we tried to track that on the SecSpider website. It's just sort of best-efforts. So, I think in the past people have pointed out we were missing some zones which we've tried to catch up on. But that's one place to find some aspects of growth.

STEVE CROCKER: Good. And there's a question. "Can we post links of the projects that Eric and the team are working on? I'd like to follow up." Well, the answer is yes, we can post them. The question of where would we post them? Let me just hold off on the rest of that and let Dan York chime in here.

DAN YORK: So, yes. I don't have specific statistics broken down by region. I mean, we have the maps and we have some data in there that show kind of where the DNSSEC—what the signing is of the ccTLDs. But that doesn't show, inside of those ccTLDs, what the adoption rates are inside of that. So, I'm not quite sure who does because each ccTLD … Or, I mean each … Well, it's challenging to measure some of that. Geoff Huston at APNIC breaks down validation. So, the checking of the signatures. And so, we're seeing that measure on there. And you can get it broken down by region, etc. So, that would be the answer there.

ERIC OSTERWEIL: Yeah. I could jump in on that, too. So, I probably misunderstood the question and so Dan sort of hit me with a clue about it which is nice. Thanks, Dan.

But we have a bunch of delegations that we track below the TLD level in SecSpider as well. We track everything that we can find. And we have … I think it's a pretty cool plot that takes just the top biggest TLDs and then plots out how big their islands of security are below them, whether or not everything is delegated or not. But also, we have this thing called a stream plot which is really cool, and you can actually go and look at how deeply we can find …

We found zones under each of these TLDs. And we can do that for anything. But we just do it for the top 10. So you can go to the SecSpider page, and if you go and look at the Hierarchy tab, you'll see a stream plot down below it. You can mouse over and see that some zones have hundreds of zones at the 12 LD and deeper, and stuff like that. So, that's one thing that we do, and I'm happy to share data if anyone's curious about anything.

DAN YORK: I'll also mentioned that the DNSSEC tools site that the stats.dnssec-tools.org that Wes Hardaker and Viktor Dukhovni and others have been working on. They do have a page on there, or a site on that, breaking down, again, on the TLD basis. They're gathering it in there. So, there is some of that but it doesn't, again, roll it up into regions.

JACQUES LATOUR: I had a question, but I can't post it in the Q&A because I'm a moderator, I guess.

STEVE CROCKER: Speak.

JACQUES LATOUR: So, if a child publishes a CDS or a CDNSKEY record and a registry can pull it, a registrar can pull, a DNS operator can pull it, is there a standard to know who's on the hook [inaudible] different information? Or it doesn't really matter because if the CDS is properly signed and it's legit, then it should make its way to that registry's DS record.

STEVE CROCKER: So, Jacques, I attempted to actually answer that in the chat. My view—and this is just simply my personal view about estimating the way this is likely to roll out—is that it's the registry that decides whether they're going to do it or whether they're going to leave it to the registrars. And they have to make that obviously known to everybody. So, if they're not going to do it, then that becomes obligatory for the registrar or for some other means to be done. And if they are going to do it, then all the registrars know that they don't have to do it. But as you point out, there is room for some confusion. And the potential good news is that if both parties do it, the result might turn out to be the same.

JACQUES LATOUR: I think Perhaps we should modify the DPS or TLDs operators to have a DNSSEC practice statement to show what our policy is around CDS/CDNSKEY automation.

STEVE CROCKER: And the practice statement that you want to modify is the registries practice statement?

JACQUES LATOUR: That's right. It's our DNSSEC practice statement. It [shows] how we do DNSSEC key rollover and all that. I'm thinking we should add a section. So, there's an RFC somewhere that defines what should go in there, so we should modify that and address CDS/CDNSKEY.

STEVE CROCKER: So, I think that's an excellent suggestion, and I think it fits right in with the addition that Dan is making to the maps. So, Dan, maybe we can make it an implicit requirement that when there is a report that a registry is now automating the DS updates, that they also provide some reference to their practice statement that documents that.

DAN YORK: Okay. I don't know what to do with that, though.

ULRICH WISSER: I have another question for Dan.

DAN YORK: Just in that regard. I mean, yes, when we find out or if they alert us to the fact that they're now doing DS automation, it would be great if they'd provide their statement or something. But I don't know where we'd put that.

STEVE CROCKER: Oh, well I have an idea. Since you have the [inaudible] open, it's just a small matter of programming to add it to the database.

DAN YORK: Yeah, sure.

ULRICH WISSER: But, are you actually going to track support for CSYNC, too?

DAN YORK: I was not planning on being … I was planning on keeping it very simple. Do they do some implementation and turn the flag? Yes. Not getting into the specifics.

ULRICH WISSER: Yeah. Well, with CSYNC you can do the automation. Without CSYNC, you can’t automate it fully.

STEVE CROCKER: We've opened up a small can of worms here which is, I think, ultimately constructive. But I was being slightly facetious in suggesting the Dan should do more work on the database. But I think, Jacques, your idea is really quite appropriate, and rather than imposing a particular method or standard, the way to get started is for you and for others to begin publishing. And then when there is a sufficient number which might be only two or three, but whatever a sufficient number is, to start socializing that. And, hopefully, others will fall in. And then if there's some need for formal standardization of that part, it can follow afterwards.

JACQUES LATOUR: So, [inaudible], I think every TLD should have a DPS as a standard. So, maybe we should do some research to figure out what's the gap there.

STEVE CROCKER: Yep.

JACQUES LATOUR: And then we need to address that. But every ccTLD/TLD operator should have DPS.

STEVE CROCKER: Sure.

SHUMON HUQUE: I totally agree, Jacques. They should have. But my question was how many actually do have them?

JACQUES LATOUR: I don't know. I think we should track for sure.

ULRICH WISSER: And the other question where you can follow the project. There's actually … On GitHub, they've started a GitHub organization there for the project where you can follow us. And if you think that any information is missing, please feel free to contact us so that we know what to put there.

SHUMON HUQUE: And then should we also mention … There's a mailing list that Steve manages, so I think that's open to subscription by any party. Right? So, maybe Steve can elaborate a little bit on what's going on there.

STEVE CROCKER: I will type it in.

JACQUES LATOUR: And then to answer [Victor's] question. Ideally, we should be tracking the presence of a document, the DNSSEC practice statement. But there's also the tracking of the elements inside those because it states what a key rollover period … And there are parameters for DNSSEC in those documents that might be out of date based on what they're actually implemented. That could be an interesting project.

STEVE CROCKER: Good. Thank you. Excellent discussion.

It's time to move on to Mark Elkins. Mark, I'm going to queue up your slides here, I think. I don't see what I'm supposed to be seeing here. Let me try it again. Apologies. All right. Why is that … Well, here we go. All right. And what I want to do. All right. Is this visible?

MARK ELKINS: That's fine. Thank you.

STEVE CROCKER: The floor is yours.

MARK ELKINS: Thank you very much, Steve. Okay, so this is me wearing the hat of a registrar for this particular talk. And so, as a registrar, how can DNS management work for DNS providers? Next slide.

Four months ago, at the previous ICANN, I did a very similar talk. And that was with me wearing a slightly different hat, and that was gathering DS from the children of a zone. I happen to manage the edu.za zone. It's a non-EPP web-based system, etc. And it's a small zone.

One thing that's different about South Africa to perhaps too many other ccTLDs is the .za said a manager only runs .za. And different entities run the second levels. So, edu.za is a small zone, 160-odd domains. That is run independently from anything else.

What I can say from four months ago is that, well, we've had no new signed domains, unfortunately. We have had 10 more domains added, so it has grown a little bit. And what's really nice is that now there's also no bad name server records. But that was last time. So, if we go to the next slide.

So, context of this particular presentation. I am a non-ICANN accredited registrar. I am accredited for ZA and some other ccTLDs, but I can also talk EPP through DNS Africa which means I can talk directly to .com, .org, and a bunch of other ccTLDs and gTLDs. I also use a couple of other resellers as well.

That gives me a scope of being able to get to 720 possible TLDs. And personally, I'm looking after 1,500-odd domains. And of those 1,500-ish, about 600 I run. Therefore, any DS updates, DNSSEC updates, I can do immediately. It's in my system. I'm running the DNS.

But there's a good 950 domains which are being run by other DNS operators. In other words, I am strictly the registrar for those domains. I don't control the DNS. And that is what we're looking at trying to manage. Next slide, please.

So, at the moment a person can come on to my web system and, where the domain is registered, they can … If they are doing their own DNS, they can simply go in and they can push a button, re-read the [zoning code]. And that would simply pick up CDS or DNSKEY records with CDS/CDNSKEYs having the priority. And the information that would be stored in my database, and I would then simply update from there. Next, please.

What happens now is a reseller or a registrant can now come in. Without having to do anything on my side, they can simply sign their domains and include the correct CDS records in the zone. So, for example, if they're running the domain on CloudPlayer or something, then it will simply work. They go into CloudPlayer, tell CloudPlayer to sign the zone, and it works.

So, newly signed domains. The trust chain is completed after three days. In other words, after three days of me polling the CDS records, if they haven't changed, etc., and it's TCP call rather than a UDP call to check the name servers after three days of no changes, then that information is put up to the parent. And if it is already signed, then a signed domain is acted upon daily as soon as I see that. Next, please.

So, the magic at this point is simply that a PHP script runs in the early hours of the morning, digs for the remote CDS records, and basically it simply works. The complications in this at the moment is the CDS [null] domain. That's not quite … Well, the [null] domain. Otherwise, what I try and do is I try and mirror CDSs that I find to DS records and the parent. And, as I've already stated, new CDS records are recorded, and as long as there are no changes after three days, [are] deemed to be valid and therefore the zone gets signed. So, not terribly complicated. Next slide please.

So, I use dig or the PHP equivalent of a dig command to find these names server nominates. The biggest problem that I actually find is that name servers themselves are incorrect. It would seem that in the population area that I live in, people are just not keeping name servers up to date. They're adding their own name servers, etc., and playing around. And then that information is not being passed up so that I can put it into the parent.

What I can say—and I've been running this software for a while—is that I now have 180 new domains where DNSSEC has been added in the past three months. And a number of resellers have actually fixed their name servers, so the information now sitting in the parent looking at the children is more correct than it has been, basically, ever. And people are actually using this to sign their domains with DNSSEC which, to me, is the great thing. Go to the next slide, please.

So, when the program runs at 3:30 in the morning, it generates a log file. It works customer by customer or reseller by reseller. So, I'm just showing my own information here. I've got two domains that were not on my equipment, so from an experimental point of view we can see what was going on. And I managed to catch where the domain smtp.co.za actually activated after a few days or up to three days. The fact that there was a DS record and it copied it up at this particular point into the parents. And the log file is, like I said, about 630 lines. And 947 domains later, we see that this took 11 minutes-odd to process. For the 947 domains, you have an idea of the speed. And there were three zone changes.

But this really is a work in progress. The fact that people simply have bad name servers is one of the worst things at the moment. I also see that a lot of name servers are unable to … You can't ask them about CDS records. They'll come back with an error. So, people's software is also not necessarily up to date. Next slide.

And that is it. Questions, if any? Back to you, Steve.

KATHY SCHNITT: Steve, you're muted.

STEVE CROCKER: Thank you, Kathy. Thank you, everybody. I made a point of imposing very strict time limits on the people on my panel, and everybody cooperated. So, I'm very grateful. I apologize for any glitches that I inserted.

It would be very, very helpful to have feedback. This is, as we said, a continuing series. And so, the pattern that you see is the one that we've evolved, and I'm happy to adjust it to improve on it if we can. And for lack of feedback, we'll continue doing the same thing, so be warned.

But we will return with the willingness of the program committee, which has been very generous to us, as a continuation of this with the same topics, and hopefully be able to report substantial progress along each of the paths that we're on.

We have a couple of minutes for any additional discussion questions, suggestions, etc. But the other alternative is to actually return a few minutes to each of you for extending the break that's coming up. And for those of us in the Western hemisphere, that means lunch. Oh, boy.

ERIC OSTERWEIL: Steve, can I make a plug? Just since we have a little bit of airtime, I would love for people to continue to volunteer zones to SecSpider's database if that's possible. If you're interested. Just hit me up if that didn't come across as clear to anyone.

STEVE CROCKER: Good. And Jacques, you've been lit up. Were you about to say something?

JACQUES LATOUR: I'm hungry.

STEVE CROCKER: You’re hungry. All right. Kathy, let's declare success. We're going to reconvene in 40 minutes.

KATHY SCHNITT: Beautiful. Yes, we will reconvene at 17:30 UTC. So, thank you, Steve. Fabulous job. And thank you to all the presenters. Please, stop the recording.

[END OF TRANSCRIPTION]