ICANN73 | Virtual Community Forum – DNSSEC and Security Workshop (2 of 3)
Wednesday, March 9, 2022 – 14:30 to 16:00 AST

KATHY SCHNITT: Hello and welcome to the DNSSEC and Security Workshop Part 2 of 3. My name is Kathy and I am joined by my colleague, Kim. And we are the remote participation managers for this session.

Please note that this session is being recorded and governed by the ICANN Expected Standards of Behavior. During this 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 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 and speak at that time.

All participants in the session may make comments in the chat. Please do use the drop-down menu in the chat pod and select Respond to All Participants and Attendees. This will allow everyone to view your comment. Please note that private chats are only possible among panelists in the Zoom Webinar format. Any message sent by a panelist or a standard attendee to another standard attendee will also be seen by the session host, co-hosts, and other panelists.

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

toolbar. To ensure transparency of participation in ICANN's multistakeholder model, we ask that you sign into Zoom sessions using your full name. For example, your first name and your last name or surname. You may be removed from the session if you do not sign in using your full name.

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

STEVE CROCKER: Thank you, Kathy. And thank you, Kim. Welcome to this ongoing series of panels. We're now in what I call Episode 7. So this means at three per year, we're now in our third year of running these.

What is it? This is focused on a particular set of rough edges, if you will, in the DNSSEC specifications and implementation and practices that have to do with the provisioning process. And in particular, two specific things. One is automation of the movement or creation of DS records in the parent zone. And the other is the mechanics of coordinating either the transfer of an operating domain, an assigned domain, from one operator to another or, more generally, bringing two or more operators into cooperative service over the same zone.

Shumon Huque and I have been chairing these panels for, as I say, the past two plus years. And the work is not yet done, by a fair amount, and so I think we'll probably continue for maybe even at least another year and perhaps more.

But that said, there is great progress. In particular, this session is quite exciting because of a handful of things that I will mention here. So a very

quick review. As I said, there are sort of gaps in the protocol specs—one on the automation of DNS update and the other having to do with the integration or coordination of multiple DNS providers.

These are background slides. I've used these before. I'm going to go very quickly past these. Not at a tutorial level. All of the slides are available online. And of course, you're welcome to ask questions toward the end.

We have, in addition to this overview that I'm giving, seven other talks. Brian Dickson from GoDaddy, Ullrich Wisser from Swedish Internet Foundation, Nils Wisiol from deSEC Technische Universität of Berlin. You wouldn't know that I actually studied German. But did not do well.

I will introduce a new piece of work within SSAC. Johan Stenstam from the Swedish Internet Foundation talking about the multi-signer software called MUSIC. Shumon talking about some adjustments that are needed in the formal specifications in the RFCs related to dealing with multiple algorithms. This touches on, in a way, with things that were mentioned in the last session.

And then finally, Pouyan talking about work on observatory of watching changes over time in these systems. So that's the agenda.

For DS updates. The issue, as I said, is that when there's been a change in the child zone—a key rollover and re-signing of records—there has to be a corresponding change in the DS key up at the parent level. In a very large number of cases, the registrar is also providing DNS service. In which case if they're providing assigned DNS service that is with

DNSSEC, they have the access to the registry—to the parent zone—and can simply push a DNS record up.

The problem comes if the DNS service is not being provided by the registrar and is being provided by some other party, either a third-party commercial operator or, internally, by the registrant within their organization.

The only official pathways that were defined are through the registrar copying the cryptographic keying information by hand into a web page. It's a very awkward, slow, and error-prone process. So the question is, can that be automated in a sensible way?

And the answer is yes. And there's, at least in principle, push mechanisms which are indicated by the blue arrows or pull mechanisms. That is, pulling from the top and looking inside of the zone from the bottom. In practice, only the pulling processes have been implemented. I'm not sure I know exactly why, but that's the direction that implementations have proceeded.

We now have updated the maps that Dan York introduced at the beginning of the last session to include an additional state of whether or not the ccTLDs do in fact do this kind of polling. And a few—not many—but a few now do this. And the news that you're going to hear next is that GoDaddy is now in the early stages of doing this sort of work—polling—at the registrar level. And then they can, of course, push that data up into the registry.

This is just to emphasize that this is the same thing. You can have a registrar that is not providing the DNS service, but is the registrar record. And the registrar can do the polling to the child zone which is being operated by a different provider.

This is [not safe for an] ICANN type of map because it shows national boundaries. And hopefully they're not changing too much, even in this region. But the blue regions are where there are countries—ccTLDs, really—that are doing the scanning and looking for updates for DS records.

A question I have is whether or not the scanning process will scale properly. And that's something that we'll watch over time and see how that works.

Here's a scorecard that work is proceeding. And it's got two sides to it here. The scanning which I've mentioned, and then there is also a question of how do you initialize these relationships. And you'll hear a bit about the bootstrapping process today from Nils.

Meanwhile, over on ... How do you coordinate the operation of multiple independent providers, each of whom have their own keys but are serving up the same zone? There has to be a degree of cross-signing. And there are, sort of, two interesting scenarios. One is where you want multiple providers on a continuing basis for capacity and for reliability in case there's partial or a failure on one side.

And the other is how do you transition from one provider to another? The transition turns out to be just the limiting case of having multiple

providers. And the way you transition is that you bring the second provider up to speed and then after some amount of time which can be quite short, you phase out the original one.

But that, as I say, is a limited case of simply having multiple providers. And in principle, there could be more than two. I'm not sure we've done any testing of that, but the algorithms and protocols certainly, can work in that fashion.

There is a substantial project underway with several parties anchored at the Swedish Internet Foundation to develop multi-signer software. And you're going to hear about that, particularly from Johan today. That involves, as I say, people from each of these other organizations. And that is leading to a proof of concept to run the transition processes and the coordination processes in a fully-automated fashion.

The key thing that has to happen—pardon the pun—is that the key sets have to be cross signed. That is, the key sets in one copy of the zone that are operated by one provider I have to be signed with the keys in the other one, and vice versa. So this will include demonstrations.

As I say, I'm flipping through this very quickly in order to preserve the time for you to hear the real talks on this. But that work is going along. This is kind of a scorecard on what's done and what's in progress. You'll see along the left side the software. There's still a certain amount of work that is in progress. But key pieces are don, and more is coming along.

It also includes test beds that are being set up and experiments that will take place with those test beds. And then here are the components that go into the multi-signer controller. Here's a scorecard on implementations. The design work is in good shape. There is implementation in progress on multiple fronts, and there are also pieces that are now complete.

So PowerDNS and BIND software packages have the necessary interfaces deSEC is a service provider that Nils is from that's up and running. And meanwhile, NS1, Neustar, and Cloudflare has now announced that it, too, is in progress. And I think that we'll hear more from Cloudflare at the next panel in June.

Here's another representation of the progress. The green with the check marks are complete. The orange with little squares are in progress. And we have room to add a score for others. And if anybody is doing something that we don't know about and wants to be included in a future representation here, please do contact us.

This was on the software packages. This one is on the service providers. deSEC, NS1, Neustar, Cloudflare. And I'll just say it's now beginning to be an interesting set. I hope that we get more green going forward and more columns added as we progress.

I said we've been running these for quite a while. These are these seven sessions, including this one, with TinyURLs that point to the sessions. And then gradually of going back and filling in the table of contents, if you will, for each of the prior sessions.

So with that, I want to now proceed and turn things over to Brian Dickson from GoDaddy. And to all the panelists, as I've mentioned, I'm hitting the slides on your command.

Brian, the floor is yours.

BRIAN DICKSON: Thank you very much, Steve. That was an excellent overview. So this is who I am and what we're doing, or what we're talking about. The stuff about doing the polling. Next slide, please.

So in the overview, Steve covers all of these scenarios quite nicely. This is simply an instantiation of showing what GoDaddy currently does and supports. We always publish CDS/CDNSKEY even when we're not the registrar, if we're managing DNS. And what we're adding in is the polling of third-party DNS where we're the registrar. Next slide please.

The current state is that we've started testing. And anyone who's interested in participating in the testing is encouraged to drop me a line. The obvious requirement is that the registrar for the domain has to be GoDaddy. But [inaudible] that, we should be able to actually do the polling and the updates.

It's currently like an open-closed beta. So it's not automatically discovering domains and we're limited it to polling of participants of the beta simply for safety reasons. And then once we're happy enough with how it's behaving and once we fix any bugs, we will expect that to be pushed into production. Next slide.

This is just the summary, in table form, implemented and now being tested of GoDaddy polls the child zone of a third-party DNS operator for a domain that is registered through GoDaddy and, when appropriate, submits the EPP commands for updating the DS records or DNSKEY for certain TLDs. Next slide.

Is there a next slide? Yeah. So once we've got all that working and in production, we will also be participating and using that mechanism for doing the automated Bootstrap of the initial DS submission. And then we will also start participating in the Multi-signer support once we have this in place.

And that's pretty much my update here.

STEVE CROCKER: Thank you. Alright, that moves us along to Ulrich talking about a topic that I don't think we've talked about before having to do with the CSYNC. The floor is now yours, Ulrich.

ULRICH WISSER: Thank you, Steve. Hello, my name is Ulrich Wisser. I work for the Swedish Internet Foundation. We can take the next slide, Steve.

I think on the last meeting, we actually talked shortly about how this CSYNC record works and what it does. But now, actually, we have released it in production. So for the .se zone, we now scan actively for CSYNC records and in case there is any, we would actually update the registry. And what we support is updating NS records and possibly glue

records. The .se zone only takes glue if it's strictly necessary, so it would only be supported [inaudible]. We'll take the next slide.

It's just a short timeline. We implemented CDS scanning last year which no means that you have ... All the technical information that the registry stores about a domain can now be updated through the DNS. Next slide, please.

So what this does, actually, is if you ... It actually helps with something that we have had problems with in our industry for a very long time because in the traditional registry/registrant model, there is no place for DNS operators. They have no access to the registry. They can configure the domain in any way. But now once the domain is assigned to your name servers, actually the DNS operator can configure the domain. And they can change DS. They can change the DNSSEC parameters. They can even change the name servers for the domain.

And this is actually a big step. It allows for infrastructure updates for [hosters], and it allows us to do DNSSEC automation in the multi-signer project in the end.

We have contact with our registrars. And you would think that why is this not for registrars? They have no [use] for this? But if you think about it, a lot of our registrars are actually big hosting companies and in part of the domains that are running on the infrastructure is domain that has been registered with other registrars. And they have no possibility to administer these domains. But now they have. And so this is actually something that a lot of people are looking forward to.

And so then I have just one more slide please, Steve. And that is little statistics about our DDS scanning. So actually the CSYNC scanning is so new, we don't have statistics about it. But the CDS scanning you see there is usually ... It's about, like, three domains getting updated per day. And then we had one registrar actually using it, and we had over 3,000 domains in a single day. But, yeah, it's ...

Well, we're not overwhelmed by usage, but it's a steady usage of the system and we're really happy how it worked out. And we have had no complaints whatsoever so far.

And that's my update. Thank you.

STEVE CROCKER: Thank you very much. All right. Nils, this is the talk you're going to give. Yes? That you and Peter have been working on, automated DNSSEC bootstrapping?

NILS WISIOL: Yes, hi. Thank you, Steve. Next slide, please.

So we've seen this chart on the right before, and what we realized was that the mechanisms that we have for the DS bootstrapping either involve manual action while pushing to the top ... You can see the blue arrows on the left. So if the DNS provider is separate from the registrar, it involves the registrant doing things which is usually a bad idea, I believe.

Or if we're using CDS/CDNSKEY, then on the right it's not authenticated. And that's what Peter and I want to address in an extension to RFC 8078. We want to add authentication to the pull from the bottom. And we tried to do that by co-publishing the CDS/CDNSKEY records in the zone of the name server name. Next slide, please.

So there's an example here. Imagine you have example.com and you want to bootstrap it. So it's not yet secure and your provider is ns1.provider.net. Sorry, that's your name server. It's already standardized. You put the CDS/CDNSKEY records in the example.com zone at the apex. But you also put them in the zone NS1.provider.net. And you can see on the left with the yellow stripe in between, that's what we mean by co-publishing. It's the same information in both zones.

And now we require just the names server operator to secure the name server zone. And now the registry or the registrar can look up those records by just knowing the name, knowing the name server name, and can confirm the authenticity of this information just by using DNSSEC. Next slide, please.

So, technical considerations for this proposal. There's no collision. You can think, "We, CDS/CDNSKEY is for something else," but they're only used at the apex right now and we do not propose using them at the apex in the name server zone.

And second, we do want to add an extra label because that allows us to put this whole information into a separate zone. So we don't actually want to put it in the name server name zone. We want to put it in a zone

that is below this. So you can use separate keys. You can do online signing. You don't need to mess with your sensitive zone and you don't need to be afraid that it will break your operation. Right?

It also allows to use XFR, if you would like to use that. So the registry or the registrar could get all your CDS records at once. And that's why we propose to put it in a separate zone. Next slide, please.

The question is, is this useful because it requires the name server to be in a secure zone? And we think it is based on the fact that in the Tranco Top 1 Million, 25% of zones do have secure name servers. Most of them are Cloudflare, by the way. But the zone itself is not securely delegated yet. So we believe there is potential for using this at least among the Tranco Trop 1 Million. Next slide, please.

What's the status of our project? We do have a prototype implementation where you can throw in the name of the zone and the name of the name server and it will spit out the DS records that are authenticated using this proposal.

We talked to CoCCA, and they want to deploy CDS and CDNs scanning for their 59 ccTLDs plus using the authenticated bootstrapping. Brian mentioned earlier that GoDaddy is also planning to use this. And we talked to .cl. They finished implementation and waiting for approval.

And the IETF DNS workgroup told us that an adoption of the corresponding draft is in sight, so we hope that will happen during the meeting next week, possibly. And we published a blog posts at APNIC to get the word out to push people to adopting this. Next slide, please.

Oh, that's it. Thank you. That was our update.

STEVE CROCKER: So I'll just flip through these in case you want to spend any time on them. But these are part of what's posted. Thank you, Nils.

All right, my turn. A very short presentation. The presentations that you've seen so far are focused on actual, real work that's underway. Within the Security and Stability Advisory Committee, we have now started what we call a work party. The aim of this is to generate a report with recommendation and how is that related to what we're doing here.

So we're taking note of the work that's underway on the DS automation, and want to publicize it on the one hand, raise awareness. And make recommendations that address registry operators, registrar operators, third-party DNS signers and operators, and the software vendors that support all of these operations to suggest, in an organized way, that interfaces ought to be part of the packages of the software packages and that the procedures and processes that the operators run should include the automation process and.

These recommendations, the intent is ... We've just started this work. We're not anywhere close to being done yet, but ideally these recommendations will speak both to the entire Internet community and then also more specifically within the ICANN gTLD contracted party community with the thought that maybe there would be some organized adoption and even contractual requirements. But that remains to be taken up. But that's the direction of all of this.

And just to repeat, so the main goal is to remove provisioning roadblocks and facilitate the deployment of DNSSEC operation, and specifically with respect to the things that are listed there below. I think that's ...

We will be doing some outreach with the different groups, particularly in the ICANN community, both to share what we're thinking about and also to get feedback and incorporate that in any draft recommendations that we put together.

Nominal timeframe is to be completely done with all of this this time next year, hopefully even sooner. But probably longer than it will take to be completely done by the Annual General Meeting which comes early this year in September. So somewhere in the fall, I would hope that we would have this pretty well under control. We will see. And we will report at least briefly during these sessions in the upcoming meetings.

So that's that, and happy to entertain questions. We'll do a broad Q&A at the end of all of this, so put your questions in the Q&A panel. There's a chat panel and a Q&A panel. Best to put your questions in the Q&A panel and we'll get to all of those.

So with that, let me move on to Johan Stenstam talking about the software development that they call MUSIC.

JOHAN STENSTAM: Thank you, Steve. Can you hear me?

STEVE CROCKER: Yes.

JOHAN STENSTAM: Good. So this is a presentation of what we've been doing recently at the Swedish Internet Foundation. And obviously it's sort of different from what others have been doing because they have sort of been making the groundwork and opening up API's for things to be managed from elsewhere, from a third-party controller like the one we are designing, etc., while we are sort of doing the easy part of just bringing it together, given that everyone else opens up their stuff.

So with that said, let's go through the first few slides rather quickly. Next slide. Next slide. And next slide. And next slide.

So as all of you know, we're playing with this game the DNSSEC game, for many, many years in some cases. And doing these decades, lots of progress has been made. So compared to the initial versions, we are in really good shape without getting into all the details. But now DNSSEC is well known and DNSSEC knowledge is widespread.

But we still have some tricky corners here that are still causing us problems to reach, really, wide scale adoption and also wide scale comfort with DNSSEC. And the comfort part is really part of the problem here because it's one thing that you make something work from a technical point of view. It's a different thing having people being comfortable with it, not being concerned that it breaks [their own time], etc. Next slide, please.

So if we focus on the difficult parts here, skipping past the first one which is debugging stuff when it breaks, and we look at the case of communicating with the parent. And this is clearly just a lead-in to what others have also been saying about CDS, etc. Next slide, please.

We really want to automate the parent-child interaction here. The interaction typically on the child side being managed by DNS operator these days, and the parent ... Well, the parent being the parent. Next slide, please.

And the answer to all our problems is in the DS case for the DNSSEC, part of the delegation information. The answer is CDS. And for the traditional delegation information the answer, we believe, is CSYNC. And implementations are there, and adoption is spreading, although from a low base. Next slide, please.

So from my point of view, having played in this space for, well, 25 years now, I've always been astonished by the fact that we seem to be unable to get away from a certain amount of delegations that just are not right; delegations that are not quite in synch between parent and child. And the reasons are obvious. You have to maintain the same stuff in multiple places. And that is always difficult. And it's always [inaudible]. And when it still works, you sort of defer that and it doesn't happen.

And now, finally, we are at the point where we can automate this. And automation is obviously the way to deal with things that are boring, things that are complex, and things that are dangerous to tweak with. So suddenly we go from a point where DNSSEC is adding complexity to

a situation where DNSSEC is actually sort of removing complexity, is removing a difficult [chore]. And that is making everything much better. Next slide, please. And next slide.

So, given that if we go back to the progress and failures here and we ... Go to the next slide. The next difficult corner here, which is really what was the core focus of my talk is, is the problem of changing DNS operators. And Steve already introduced this and it has been talked about in previous meetings, etc. So this is clearly where our attention is and where we consider the crucial problem to solve, to [reside]. Next slide, please.

Why is it difficult? Well, you can look at that from several different vantage points, but one way of looking at it is that it's simply difficult. It's a number of steps. It involves cryptographic keys. It involves parties that have different business incentives to help. It's an industry with paper-thin margins where it's really difficult to spend effort on a couple of zones rather than all of the zones because as soon as you just touch a single zone, your revenue on that particular customer is gone forever.

So it's a combination of factors that make this a difficult thing. And obviously, when there's not enough incentive to do it manually and there is not enough revenue to do it manually, you need to automate. Next slide, please.

So the MUSIC software that we've been developing for some months now is obviously based on the so-called multi-signer draft by Shumon and Ulrich. And in that draft, they describe processes to migrate from a

signed servers with one signer, or one set of signers, to a new set designers where you either add or remove signers.

From our point of view, a signer is a service. So a signer is something that can take an unsigned DNS zone, generate whatever keys or necessary, sign it with those keys, and in the end publish that signed version of the zone. When we have multiple signers, they generate their own keys.

And what MUSIC does, what the multi-signer controller does is to talk to all signers and fetch information from them and analyze the information and then make sure that each signer has enough information about the data from the other signers to make everything work without any interruption, without going unsigned.

So the most common case, or the poster-child case, which Steve mentioned is going from one signer through a temporary state of having two signers, and then going back to having a single signer being the newly-added one. And that would describe the process of essentially changing DNS operators.

The other case, or one other case, is to go from a single signer to multiple signers, like A and B, and then stay in that mode for perpetuity because you want that from a resiliency point of view or robustness point of view.

And the way that MUSIC works—and this is something that we will look further on in the next slides—is through a series of steps. And I cannot emphasize enough that it's a complex process with several steps, lots

of checks and balances. But the way the processes are designed, every single step is safe.

So there are no timing constraints anywhere. There are no steps where you have to be sort of quick and not break for lunch in the middle of it. Every single step is safe. Every single step is somewhere you can pause for one minute or one month or forever. It wouldn't break anything. And that the multi-signer logic solves those problems is really the core for this to be successful. Next, please.

So, what does MUSIC do? Well, to begin with it looks at the DNS key RRsets from each signer, analyzes that, extracts the so-called zone signing keys, and adds the respective zone signing keys from one signer to the other signer which is needed for the DNSSEC signature change to be complete. Next slide, please.

In addition, MUSIC adds CDS records to each signer version of the zone. And these CDS records will cause the parent to—given some time delays, etc.—updated DS records in the parent zone. Next slide, please.

And finally, MUSIC will update the delegation information in the parent through the addition of CSYNC records, also through the various signers which—again, after [sometime later]—will cause the parent, having scanned and seeing the CSYNC record, update NS records and potentially also some glue. Next slide, please.

So here we have a process. This is the add-signer process. As you can see, there's a significant number of steps here. So each box is a state. A state is, well, it's a state of which a zone can be in. The zone can be in a

state where the signers are unsynced. That's the initial state. Or the zone can be in a state where the DNS keys are synced, but not the rest of the steps have been done, etc. There's a bunch of different steps, all of which are valid. Next slide, please.

So between the steps—I don't know if this is readable or not—there's something called an action. So action is a transition. The change from one state to the next state is described by the action, what has to happen for the next state to be reached.

Actions are not initiated randomly. They are only initiated if certain preconditions are fulfilled. And then the action happens. And then after the action has happened, there is a set of postconditions. And if the preconditions fail, there will be no action. If the postconditions fail, we will basically roll back and the action will be undone.

So both preconditions and postconditions will have to be successful for a state transition to take place. And that in combination with every single state being fully valid means that we can regard this as a safe process. Regardless of what happens during the process, wherever we have to stop for long or short, this will work.

The remove-signer process is obviously similar to the add-signer process except a certain number of states are in the reverse order. Next slide, please.

So from a design point of view, MUSIC is based around three concepts. There are signers, as in entities that can take on signed zones, generate

[inaudible] keys, sign these zones with keys, and publish them. There are zones, obviously.

We don't care about how the zone contents are maintained. We don't care about how the zone is kept in synch with the signers. That's outside of scope. We only care about being able to talk to designers to extract information and talk to designers to insert new information to make them get in sync.

Then there's an obstruction called the signergroup. And the signergroup is basically a container. And in this container, you have a bunch of zones and you have one or more signers. And the idea with the signergroup is that, let's say that you have five zones—or let's say that you have 5,000 cells—presumably, you will use basically the same set of signers; as in, you will use the same DNS provider or you want to move all your zones from DNS provider A to DNS provider B.

What you do is add the zones to signergroup and then you add a signer to the signergroup. And whenever you add or remove a signer from a signergroup, all the zones in that group will automatically go through the appropriate MUSIC process—be it add-signer if you added a signer, or remove-signer if you removed a signer. Next slide, please.

So, that's all nice, saying you make changes. But how does that work in practice? Well, this a third-party controller and for the third party controller to be able to make changes to production zones, obviously we need authentication and we need security and we need all sorts of authorization for that to work.

The first mechanism that has been implemented is standard DNS Dynamic Updates. That's well-known technology. We know how the authentication works. We know how the authorization works. And there is also support in more than one open-source name server implementation, which helps. So that works fine.

The second alternative is the various commercial or free DNS services that provide various types of APIs as the access mechanism for making changes. In our case we're using the deSEC API as the proof-of-concept. But in no way is the MUSIC design restricted to that particular API.

We don't care, basically, how the API looks as long as it supports the functionality we need. It's a plugin-based design, and as long as you design a new plugin for some new API from some new provider that supports the three operations we need, we're happy. Next slide, please.

So what is the state of this and what is the goal of this? Well, this is not an industrial- grade implementation. I want to be very clear about that. This is a proof-of-concept thing. It’s really cool. It’s really useful. It works fine. But it's still a proof-of-concept. And we wanted to prove that it was possible to do this operation of migrating from one DNS provider to another one for a DNSSEC signed zone without ever going unsigned and without causing problems during the transition. And we are fulfilling that goal.

It's a server-client design, so there is a secure API between the client and the server. The server does all the maintenance of all the steps. It tracks all time constraints, tries again whether it can move a zone to the next state. If not, it waits for a bit and tries again. And it tracks all these

details. It tracks all these preconditions and all these postconditions for every single zone and every single signer and does a lot of work in the background.

The client today it's just a command-line client time. It could be anything. It could be a web frontend. It could be part of a larger provisioning system, but what we've done is a CLI tool implementing our API for talking to the MUSIC server. And I've already said that MUSIC is rather safe. Next slide, please.

So here we have an example of me running this. It's not a complete example. It's just showing a couple of things. So the first one in the first blue box is me adding a new zone to the system with the command "music-cli zone add". And the -z flag is followed by the name of the zone. And the -g flag is followed by the name of a signergroup.

And as you can see, the output is that the zone was added. So that's good. We can list what zones are in the system. And the -H is just a flag to get headers on the various columns.

As you can see, in this case we have three zones under management. They are in two different signergroups. A signergroup is a very cheap obstruction. You can have as many signer groups as you want, if you have different constellations of zones that should be managed by different DNS operators, for instance.

So we have two signergroups, G1 and G2. We have one zone, the test1.zone which is apparently in sync, and MUSIC is happy about it. We have the new zone that we just added, and that one is apparently in the

add-signer process which is what MUSIC does for a new zone because it doesn't know what the state is when the zone is added.

So just to be safe, it goes through the add-signer process with the current signers for that signergroup. And the first state in the add-signer process is the signers unsynced state. So here we have already progressed from the first state to the next state where the DNS keys are already in sync. And over in the right-most column, Next State, you can see that the next state after this will be the additional CDS record.

And then we have another zone called other1.tld here, which is in another signergroup. And if you look carefully, you'll see that that one has been sitting since February. So that has apparently ceased propagating for various reasons. Never mind. It's just there as an additional example from a different signergroup. We wait some more and then we look again, listing the zones.

And now the test2.zone has come in sync. So the MUSIC server has done its thing without the command-line tool having to interact in any way. It has pushed it step by step through the process, and whatever signers are in the signergroup G1 or now in sync. Next slide, please.

So now that everything is nice and dandy, let's try what we really want to try here, and that is adding a signer or removing a signer. In this case we add a signer. So we use the commend “signer add -s (name of signer)” which is just a name.

And then there is a whole bunch of arguments when you add a signer. There's a method. That’s “-m ddns” in this case because it's using

Dynamic Update. And then there is a bunch of IP addresses. There is authentication details and secret keys and all sorts of details. For reasons of brevity and making the slide comprehensible, I used ...

I pushed all that stuff into a file. And “cat foo.details” just means all the details that describe the signer foo. And “please add it to the signergroup G1.” And the output is that it was added and it was attached to the signergroup G1.

What should happen now is that all the zones in this signergroup should immediately go through the add-signer process because we [inaudible] signer. And if we list the zones again, we see that test1.zone and test2.zone are again in the add-signer process, and they are propagating through it and they've made it part of the way through the “dnskeys-synched” state. And the next state will, again, be “cds-added”.

We wait a bit more. And if you look at the timestamps, you can see that this is a couple of hours later. And now they are in synch. The reason why it takes more than a trivial amount of time is because we have to wait for the DS record in the parent to propagate, and that takes some time.

But apart from that, it’s sort of self-maintaining. If you add the zone, it will take care of the zone. If you add a signer, it will take care of whatever has to happen with the zones in the signergroup when this signer was added. If you remove a signer, the same thing happens.

MUSIC tracks all the timestamps. MUSIC tracks all the states. MUSIC tracks all the preconditions and all the postconditions to make sure that the zone [inaudible] form and the shape we want it. Next slide, please.

So we are we're basically done from the point of view of proving what we wanted to prove. The question now is, how useful is this? Are there more use cases? Well, we already discussed, in addition to moving from one DNS operator to the next DNS operator, also being able to basically run in a prolonged fashion having multiple operators. So we can do that, clearly.

Another case is that you could be using a publication pipeline dependent on one HSM for key management. And you want to, in a controlled and easy fashion, migrate to a new HSM. Now and then you have to replace your HSMs. And either you do this manually and you're really, really, really careful, or you possibly use MUSIC or similar software to automate this process.

Another alternative could be that you have outsourced your zone signing and you want to have multiple operators. This previously was really, really hard. And now it it's still really, really hard because MUSIC is not production software, but we are proving that it could become much easier. Next slide, please.

And that's basically where we are. The system is open source. It's available on GitHub, and you will get the link. We have things we want to fix. In particular, we really want to fix the last remaining steps for

making deSEC API work. That's my fault. I've been too busy. But it's really easy and we're very close to making it work.

We would like to create some other user interface than just a CLI tool. The CLI tool is working nicely and it does its job, but it's clearly not the best presentation tool for complex information.

We have written a simple CDS/CSYNC scanner for testing purposes. We needed that to test and debug MUSIC. That is not part of MUSIC, but we still want to clean it up. And then I'm sure there's a bug somewhere. Next slide, please.

That's where we are. I'm happy to take questions now or later.

KATHY SCHNITT: Steve, you're muted.

STEVE CROCKER: There we go. We have a Q&A session setup at the end of this whole set of talks. We've got two more talks. Put your questions into the Q&A panel, if you will. I think we're doing well on time. We'll have time for a substantial set of back-and-forth when we finish, as I say, two more talks.

Thank you very much, Johan. This is a critical and major piece of work. Lots more to be done, obviously, but you've gotten to the point where we're now getting some action. And people who want to engage, if I might say so, are welcome to contact you or others and say, "How do I get involved?" Thank you.

All right. Now I've got to get back to here. Okay, Shumon. RFC adjustments.

SHUMON HUQUE: Sure. Can I be heard clearly?

STEVE CROCKER: You can. You are.

SHUMON HUQUE: Great. Thank you, Steve. And I also want to echo your comments about Johan's presentation and work. I'm really looking forward to using your implementation.

Okay, so let's get started. I will talk about some limitations in the current DNS protocol specifications that pose obstacles for certain configurations of multi-signer operation. And I'll also argue that these limitations are actually unnecessary and can be fixed with some tweaks to the protocols. Next slide, please. And the Next slide.

Alright, so let me do a quick recap of the multi-signer protocol. It allows multiple DNS providers to cooperatively serve the same DNS zone, signed with their own keys, with the use of some novel but fairly straightforward key management mechanisms.

There are two models described. But for the purposes of our panel, Model 2 is the most interesting since it also offers a solution to non-

disruptively transfer a signed zone from one provider to another. Next slide, please.

And here's a summary diagram of Model 2. And the main thing happening here is that the two DNS providers shown cross import their ZSK public keys into their respective signed DNSKEY RRsets. And the DS RRset in the parent zone above them references each provider's KSK. Next slide.

Alright, so what's the challenge? If DNS providers are using different signing algorithms, they cannot today participate in a multi-signer configuration. And as a corollary, this also means that we are prevented from using the protocol to transfer signed zones between providers that employ different algorithms, which I think is very unfortunate and that we feel is an operational gap that should be corrected and closed.

We expect the presence of operators that support disjoint algorithm sets increasingly over time simply because there will be more algorithms. Right? So there are already variants of RSA, of Elliptic Curve, and a looming generation of post-quantum algorithms that may be in our future. Next slide.

Okay, so here is one of the critical impediments from one of the base DNSSEC specifications, RFC 4035, where Section 2.2 says that there MUST be an RRSIG for each RRset using at least one DNSKEY of each algorithm in the zone apex keyset. Next slide.

And there's no way to satisfy this requirement in a multi-signer DNSSEC configuration if the providers only support differing algorithms like RSA

at Provider A and ECDSA on Provider B. And this is a real situation, by the way, that I've encountered in the field. It's not just theoretical Next slide, please.

And now in RFC 6840 we find the following clarifying statement about the earlier requirement, namely that this requirement applies to servers—that is, authoritative servers—and not validators. And that validators should accept any valid authentication path they can construct. Next slide.

And my assertion is that, arguably, these two statements are not really consistent with each other. If you think about it, if validators should accept any valid path, why should signers be required to sign zone data with one of each algorithm in the DNSKEY or DS RRset? Next slide, please.

And I said "arguably inconsistent" on the last slide because there appears to be a slippery clause here. And that's the use of "should not". So this is a weaker statement than "must not" and it means that some validators could choose to ignore that recommendation and insist that all algorithms validate and still remain compliant with the specification. The Next slide.

So the simplest proposal to fix this would be to just remove the requirement in RFC 4035 and also adjust the language in 6840 and change the "should not" to a "must not". But before going down that path, let's first flip the coin and talk about why we want may want to maintain the current rules about multiple algorithms. Next slide.

Now the first rationale is kind of an obvious one. If we require signing by all algorithms, it allows us to support the widest range of validators in the field, some of which may not have implemented support for every one of those algorithms.

So my counter here is that no serious organization would deploy only an algorithm such as a fairly new one that did not enjoy the support of a very critical mass of resolver implementations in the field. Right? So they would only deploy it in conjunction with another well-known algorithm and then sign with both of them.

And we can extend this argument very simply to a multi-signer configuration and say that no organization would choose one of the providers in a multi-signer configuration that also only supported a single algorithm that is not widely supported. That provider would also need to support another well-known algorithm and sign with both.

However, there is no need to impose the signing with all algorithms requirement across distinct providers in a multi-signer group if each of those providers supported well-known algorithms. Okay? Next slide, please.

So let's move on to what might be one of the more persuasive arguments for keeping the current behavior, and that is that requiring signing by all algorithms allows validators to detect algorithm downgrading attacks.

For example, an attacker that tried to strip away signatures of the stronger algorithms in a DNS response. But this rationale is not only not

stated anywhere in this current speciation, but actively discouraged by 6840 in the language that I just cited which says that validators should accept any single valid path.

But absent other considerations, it's probably a good thing to have algorithm downgrade protection. But in the general case, only the zone owner knows the intent of their use of multiple algorithms and whether or not they want to support algorithm downgrade protection or whether this isn't an issue for them because they are a multi-signer configuration where the providers are using distinct algorithms but have roughly the same strength; or whether it's a zone in a transitional state from one provider to a different provider which supports different algorithms, and so on and so forth.

So the bottom line is, in my view, validators should not unilaterally imposed requirements that interfere with the zone owners' actual intentions. The gap here, I think, is that there is no mechanism today that allows the zone owner to express that intent to validators. And maybe in order to do that, we need such a signaling mechanism. Next slide, please.

So there are two main options I can think of to signal this. We could do it in the DS record set. But as we all know, it doesn't have flags. So the usual approach there is what is now commonly known as a DS hack where we devise a new pseudo-algorithm, use a DS record entry for that which carries in its data field the needed signaling information.

The other obvious option is to do this in the flags of the DNSKEY records. And I'm sure we could get creative and think of more radical changes if

we wanted to. But I will leave that was an exercise for the audience for now. And let's move on to the next slide, Steve.

As for what needs to be signal, the crux is that we probably just need a binary signal, require all algorithms to sign and validate, or don't require it. Right? But we may also want to leave some room for additional flags for functionality that we may decide we want somewhere down the road. Right? Next slide.

One of co-panelists today, Ulrich Wisser, and I have been talking about this topic for quite a while now. And we are planning to write up a protocol enhancements proposal and take it to the IETF. Comments and feedback on this topic would be welcome, as well as collaborators if you want to help work on this with us.

So with that, I will stop.

STEVE CROCKER: Thank you very much. I've been monitoring the chat and Q&A. I saw a lot of substantive comments. Queue them up. Put them in the Q&A panel if you will. And after this upcoming talk, we will have an open session and deal with as many of these issues as we can.

Thank you again, Shumon. Pouyan, it's your turn.

POUYAN FUTOUHI TEHRANI: Hi, everyone. I am Pouyan. I'm a PhD candidate at Freie Universität Berlin. And this is the joint work with Eric, Thomas, and Matthias, and I'm going to present here called DNS(SEC) Views. Next slide, please.

So why we are doing this is, to secure your zone, it's fairly easy. You can always check what's happening at the authoritative name servers. And we've been actually monitoring these types of data right at the authoritative name server to our monitoring system called SecSpider.

But the problem starts when, from the perspective of users, of what they are seeing when things change at name servers. So they rely on recursive resolvers. Resolvers have different policies. Timing, caching, multiple signers, etc., are all factors that might lead to a different perspective that you might have as an end user from the others. So this is why we started building the DNSSEC Views. Next slide, please.

So the actual goal is to see how the distributed nature of DNS and its eventual consistency is observed and also affects end users. Next slide.

Our use case is multi-signer. You've heard about this in the previous talks. Basically, what makes it so attractive for our tool is that you have multiple stakeholders there and you have to have a precise timing and coordination among these stakeholders to have proper validation for a multi-signer scenario. Next slide.

So what do is allow an infrastructure operator, a zone owner to register their zones over our front end. And we let the RIPE Atlas probes—the nodes that do the heavy work for us—to do some measurements. We persist these measurements in a database, and we analyze/aggregate them and give it back to the public and provide some statistics and analysis on that. So what we do is ... Next slide, please.

If you enter a domain name, you will find the zone apex for that. We will schedule measurements for DNSKEY, DS, NS, and SOA records on a regular basis of six hours. So it's four times a day. It is executed by a set of random probes actually, at the moment, only in the U.S.

And as soon as we get the data back from these probes, they do the measurements. They make DNS queries. As soon as we get the response back, we check if it's signed. So if the resolver that they're using supports DNSSEC. If it's valid, it's signed. Then we're going to just persist it in the database. Next slide, please.

And to provide the analysis—so the focus is here on the DNSSEC—we calculate the different combination of DNSKEY sets. So just as multi-signer, they might see a different set of DNSKEYs. Not always the same. And a different set of active keys that they see actually matching. By active key, I mean the keys that are used to sign actual [inaudible] to create signatures. Next slide.

These different combinations are color coded. If you go on the website, if you enter multisigner1.com, that's a good example. You can see these different combinations, different color codes that we have. For example, here you see that the first color code only sees the signature for the NS records. It says "covering," referring to NS records. The second one [sees] for both NS and DNSKEY records, and so on and so forth. Next slide, please.

Finally, we aggregate all that together. We don't have a proper windowing function, but from everything that we have seen we then give you an analysis at the moment. We could say if it's multi-signer or

not. So for multisigner2.com, as the name implies, yeah, it's probably multi-signer. So that's the basic idea of the analysis that we want to provide here.

And finally—next slide, please—to sum it up, the reason that we have this, there's a measurable discrepancy between the records at the authoritative name servers and what the users actually see. It's important for the operators to know what the users see, specifically for DNSSEC and mission-critical operations.

And what we aim to do finally, if this project goes off, is to be able to get all that data raw data, to aggregate it together, and to provide analysis that can be used by operators and see if their deployment models are working as expected; and also find out how they can improve and figure out acceptance criteria.

That's it. Thank you.

STEVE CROCKER: Thank you very much. All right. I'm going to kill this, but we can bring it back up if we need to. This concludes the prepared presentations. We have, I believe, around 15 minutes more in the session, which is more than we've had in previous sessions for broad Q&A. Let's see. Let me ask for some help. Kathy or Kim, are you managing the Q&A and/or the chat?

KATHY SCHNITT: Well, for this one we've asked everyone to put their questions in the Q&A pod. So we just have a comment in there at the moment. If you want to ask verbally, please raise your hand. I'll be happy to unmute your microphone so that you can ask your question verbally.

STEVE CROCKER: Good. If there is a need to go to any particular set of slides to facilitate the conversation, I'd be happy to pull them up. So who wants to jump in here?

Well, we have ... Brett Carr put in some comments/questions. And there's been some back-and-forth on that. Brett, do you want to—

ULRICH WISSER: Can I just jump in here, a little?

STEVE CROCKER: Yeah. Go ahead, Ulrich.

ULRICH WISSER: So I just wanted ... I think Brett has also put something in the Q&A pod.

STEVE CROCKER: Yeah.

ULRICH WISSER: And I thought we might address them. So he proposed that IANA starts supporting CDS or CDNSKEY and CSYNC for updating TLD data. And I

would think that is a good idea. Maybe not for the starting, initializing DNSSEC. But for updating, that would probably be, actually, a good thing. And I don't know if somebody from IANA or ICANN would like to say something about this. Sure [we have] somebody [inaudible].

KATHY SCHNITT: Brett, go ahead.

STEVE CROCKER: Go ahead, Brett.

BRETT CARR: Oh, yeah. Well, I'll just clarify. What I was saying, really, was that I really like the idea of CDS records and CDNSKEY records and CSYNC records. They're excellent tools that obviously not many registries are currently supporting. But I think that we as a ccTLD and gTLD registry, we have to interact with IANA on a regular basis to update NS records, DS records, etc.

And one way, transitioning in or out large volumes of gTLDs which we've done several times now, that's quite an error-prone process to do through the root zone management portal. And so I think this would be an ideal way for IANA to lead the way in supporting CDS, CDNSKEY, and CSYNC to allow registries to interact with and to better automate that thing and that process.

STEVE CROCKER: Let me ask a question, if I might, Brett. I looked a couple of times at the rate of change in the root zone entries across all the TLDs. And my rough rule of thumb, which might be out of date, is that on the average—and this very, very broad; take this within, say, a factor of two—was one change per year per TLD. Are you telling me that ...

And I see you're from Nominet. Are you telling me that for the Nominet zones—of which the primary one is .uk, but you're also running others—are you making more frequent changes than to the entries in the root zone?

BRETT CARR: No. As a general course of action, Steve, no. But the exception to that is if we win a contract to run some gTLDs that we're transitioning from another provider or another provider wins the contract to run gTLDs and we transition them out from [us], then that involves us doing a large volume of operations in a short period of time. So for example ...

And Brian who's on here will probably know this already. But we've recently transitioned out about 30 TLDs from us to GoDaddy Registry. And to do that, we have to make a lot of changes in the root zone management portal in the web portal manually. And it's that kind of short-term, hard work that will be greatly improved by automation using CDS, DNSKEY, etc.

STEVE CROCKER: Thank you. That's very informative. Much appreciated. Let's see. I see hands up all over the place. I'll take them in the order that they show up here. So Johan and the Brian.

JOHAN STENSTAM: I think it's a brilliant idea. I see really no downside to IANA accepting CDS and CSYNC also. If there is a particular TLD that doesn't like this, all they have to do is not public a CSYNC or a CDS.

STEVE CROCKER: Well, clearly one of the things that we can do in this particular venue is we can invite IANA to speak to this question in three months' time at the next DNSSEC and Security Workshop. We'll see if they respond and have anything to say.

But the other thing that seems to me relatively straightforward is that this is early days in terms of rolling this technology out. And I would guess, not speaking for IANA but just as an observer, that they would tend to wait until the technology is quite firm and everything is in place and they've had time to think hard about it and make sure nothing goes wrong. So they might want to move in that direction, but maybe at deliberate speed, so to speak.

BRETT CARR: Steve, I have spoken to a Kim Davies about this in the past, and I understand that their current plans at IANA are to support automation but by a RESTful API and not via DNS standards.

STEVE CROCKER: Oh, that raises the different issue which is ... You may remember the slide I put up about the potential for push versus pull approaches. And the RESTful API would fall on the left side, on the push side.

From a certain distance, one can be neutral about this as long as there's a way to do it. And then trying to get everybody to do it the same way may be harder. A useful discussion to have. Definitely.

BRETT CARR: I think it would be just good for IANA to set the example of doing it the DNS way [inaudible].

STEVE CROCKER: Well, there's your cue, Brian.

BRIAN DICKSON: Yep. And I agree. The issue is mostly not average, but peak. There is really no upper-bound on the peak rate of those kinds of changes over a short period of time other than, you know, the upper limit is how many TLDs are there? But it's definitely going to be the case that the peak is going to be higher than the average.

The other issues, I think, might be related to the difference between the inherent nature of CDS and CDNSKEY. Those records would be signed by the zone owner, or zone operator. Which means there's a proof of

possession on the key. And unless the RESTful API requires that as well, that is going to be less secure.

But also, if it does require it then there's more overhead and no benefit to not using CDS. So that's a technical observation. But, yeah, I support and agree with the concept for changing the DNSKEY information associated with things.

So the question of CSYNC is a different one, I think. But I think that the use case is definitely supported.

STEVE CROCKER: Thank you. Russ Mundy.

RUSS MUNDY: As we all know, ICANN makes a fairly extensive use of public comment activities and getting inputs from the public on various and sundry things. One of the things that I believe is scheduled to come out for public comment very, very shortly is a study that was required by the 2016 IANA Stewardship Transition activity that deals with examining the management of the root zone process and root zone content, and so forth.

I have not seen it myself, but it does seem to me that this might be an opportunity to make comments of the nature we've just been hearing in this panel. I don't expect that there will be dealings with CDS or CDNS use by IANA, but this does seem like it would be a good input to the

public review of this study, at least for future consideration for IANA. Thanks.

STEVE CROCKER: That strikes me as a very, very good idea. And then in addition, as I'd said, we've just now kicked off a work party within the Security and Stability Advisory Committee DS automation. It's early days and we're still sketching out our task lists, and I think I'm now convinced that we should include in that task list interaction not only with the Registries and Registrars Stakeholder Groups and so forth, but also with IANA, with particular attention to the root zone and see what happens there. So that's great. This, at least from my point of view, has generated something that I had not focused on or thought of before. So thank you very much.

I don't see any more hands. Are there any more questions that we haven't dealt with? I am not—

KATHY SCHNITT: Nothing that I can see, Steve.

STEVE CROCKER: All right. I see some dialogue in the chat about downgrade that I frankly didn't follow, but I know that others—Shumon and Ulrich and others—have followed. Is there anything more that we need to discuss about that?

SHUMON HUQUE: No, I don't think so. I mean, there was some discussion, I think, prompted by Ed because I didn't mention downgrade attacks, which I think I addressed later on in the talk. So I think Ed and I are planning to talk about whether we need to explicitly document downgrade attack as a rationale in the specs because it's not stated there anywhere at the current time. Right? And Ed claims that that was an original, unstated rationale somewhere. Right? So we'll have a follow-up discussion about that.

STEVE CROCKER: Good. Well, we're at three minutes before the top of the hour, so we're right on time and we can wrap up. I think we're at a very exciting time. A lot of activity.

The next ICANN meeting comes relatively quickly. I think in June which is, what, three months from now. So typically what happens when there's only a short time is that there won't be much progress. But I think in this case we may have some substantial progress to report.

This session here was 90 minutes, which was longer than usual. The next one comes during I think it's called the policy meeting, the short meeting of the year. And correspondingly, it typically causes the entire DNSSEC and Security Workshop and also this panel to be much shorter. So when we reconvene, I expect that we will have a shorter but fully-packed agenda following some of these changes.

As I mentioned, Cloudflare has just announced that they're beginning to implement the multi-signer protocol. We'll hear more from them, I

hope. And the MUSIC project, I think, we will maybe turn it into a full-scale orchestra next time—or at least a band—and keep going.

And we have the observation work to make sure that we can follow all of this and see the transition. So lots of lots of stuff. Thank you, everybody. Any last words from anybody?

All right. Well let me thank you all for the cooperation in putting this together. From my point of view, it went very smoothly and I hope it served for everyone who was listening. And I see an amazing number of 70 plus people in this session, which is good.

And with that, we've come to the end of our time in Part 2 of the DNSSEC and Security Workshop. I'll turn it back over to you, Kathy.

KATHY SCHNITT: Thank you very much, Steve. Great panel. Great presentations by everyone. I appreciate it. We have now concluded Session 2. You can join us back here for session 3. Same link. You can either disconnect and come back or just hang tight with us. And we will begin at 20:30 UTC.

Please stop the recording.

[END OF TRANSCRIPTION]