ICANN75 | AGM – DNSSEC and Security Workshop (2 of 2)
Wednesday, September 21, 2022 – 15:00 to 16:00 KUL

KATHY SCHNITT: Hello, and welcome to the DNSSEC and Security Part 2 of 3. My name is Kathy, and I'm joined by my colleague, Danielle, and we are the remote participation managers for this session. Please note that this session is being recorded and is governed by the ICANN expected standards of behavior.

During this session, you can ask questions by typing them into the Q&A pod or by raising your hand in Zoom. And we take those questions during the time set by the moderator of the session.

You may access all available features for this session in the Zoom toolbar.

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

STEVE CROCKER: Thank you, Kathy, and welcome, everybody. This portion of the workshop is devoted to the automation of the provisioning of DNSSEC. We're focused on two relatively sophisticated aspects of the DNSSEC protocol that were not anticipated properly at the beginning when DNSSEC was designed and have been attended to later and are particularly related to DNS operators who are not part of registrars.

So the two aspects are the update of the DS records and the coordination between multiple independent signing DNS operators. That is, if a registrant has its DNS service provided by two or more separate DNS operators, each of whom signs the zone with their own keys, how do those keys become coordinated with each other. It's what we call the multi-signer protocol. So those are the aspects that we've been tracking.

This panel is a continuing operation. We are now in Episode 9. We've been doing at each of the DNSSEC and Security Workshops. So this means that this completes three years of these panels that my partner, Shumon, and I have been running. And we're making progress. [inaudible]. And at each of these panels, we highlight what the progress has been, and the people who are working on the different parts of this make presentations. So you have the experts themselves available to answer questions as well as tell you what they've been doing.

So that's the snapshot. So this covers basically what I've said—that there's two gaps in the protocol with respect to DNSSEC. One is the automation of DS updates, and the other is coordination when you have multiple DNS providers. And in both cases, what we're concerned with is how to automate the transitions that are necessary. It's possible to do these transitions by hand, but that's a messy, error-prone, and painful process. So the question is how to do them in an automated fashion and also how to do so in a way that both resolution and validation are preserved during these transitions so that any user who is looking up values in these zones gets the same answer that they would get even if the transition were not underway and things are proceeding.

So we have today the introduction that I'm providing here, followed by Brian Dickson of GoDaddy on GoDaddy's progress on DNSSEC scanning and then Peter Thomassen on automation of DS management and Roger Murray, who will present the work that's going on on the development of software that controls the multiple steps involved in a multi-signer process—a piece of software called MUSIC. Jan Vcelak will talk about DNSSEC at NS1. And I will come back and say a few words about testbeds and scenarios in this activity, emphasizing once again that this is the ninth in a series, which means that you should expect that, next time, there will be the tenth presentation. Because of the way these workshops and the whole ICANN meetings are organized, this is only three months since the previous one, and the next one will be roughly six months later. I would expect that, at the next one, the progress that you're seeing here will be supplemented by substantial additional steps, which we will be excited to present at that time.

So with respect to DS update process, there are multiple ways that are conceptually possible for conveying a DS key from a third party DNS provider to the parent zone, to the registry. These are divided into methods that involve polling from the top to pull the value up versus pushing the value into an API from the bottom. Of these, the ones that have actually been implemented are all on the polling side, either by the registry or by the registrar. And you're going to see in particular that GoDaddy has been implementing, on the registrar side, the polling process, which is good.

We also now show on the maps that were presented at the beginning … I don't know if you saw the maps, but the DNSSEC deployment maps show on the country-code top-level domains what the status of implementation is of DNSSEC and the most recent status … Let's see. Let me go forward since I'm talking about it. This map here shows six possible states, the last of which is called DNS Automation. And it's colored in a sort of medium blue. It shows those ccTLDs that do CDS and CDNSKEY scanning for polling up the DS record.

These maps are undergoing a transition from the Internet Society over to Eric Osterweil at George Mason University. And these maps have been produced for a very long period of time and will undergo a bit of updating.

This map, which I've taken from a few months ago, not from the current one, shows that there are four countries in Europe. This is off by one. There's some sort of bug in the count. The map is correct. Niue and Costa Rica also implement the scanning, in addition to what's shown here.

Let me go back to where we were. So these are the possible ways of polling a DS record up. And more particularly, with respect to a registrar polling it up, you have the registrar, which has access—oh, I have a very twitchy … Ugh, apologies. I have a very twitchy mouse here. You have a registrar who has access to the registry using the EPP protocol and implements a polling process into the child zone, which is operated from some DNS provider that is not under the control of the registrar. Because [if] we're under control of the registrar, then there's no issue about coordination here. But this is the part that is new in the overall ecosystem.

So GoDaddy is now testing scanning. We have a work party within the Security, Stability, and Advisory Committee that's supporting recommendations related to supporting DS automation. And there are some potential operational issues in the future to be explored. Scanning is time-consuming. There's some questions about how well it scales. We'll see all that works out as we get experience.

So here's sort of a scorecard on what the state of affairs are in terms of the specifications. And as I said, there's a lot of active work in this area.

With respect to coordination when there are multiple DNS providers and they are independently signing the zone and supporting it, how do you provide coordination of the keys across those independent operators and do it in a way that is glitch-free, as we say? Glitch-free means that there's no loss of resolution and no loss of validation when you bring on or when you remove one of those signing operators.

We have a project that involves a number of parties. The Swedish Internet Foundations is developing software called MUSIC, which you'll hear about. And then there are different participants testing the multi-signer protocol and providing observations and analysis.

So just to make it very clear what we're talking about, when we have multiple independent DNS providers, each one of which is generating keys and signing the zone and of course then publishing it through their own set of nameservers, there has to be some exchange of the keys between the signers so that, when somebody is looking up a chain of trust, it doesn't matter which way they go or in which order . They look things up. They get the right answer.

This just shows two possible signers, which would be the common case, but there could be more. There could be three or more.

And I should also mention that there are generally two reasons why there might be more than one DNS operator involved. One is simply providing a more robust service—so there's some redundancy—and the other is that, in the process of changing, if one is changing a choice of operator, then you have to go through a process in which of the operators, the older one and the newer one, are in operation at the same time. So that's kind of the limiting case of having two or more operators. And it's the same protocol either way. It's the same protocol if you want both of them to remain in continuous operation. Or if you want to replace one with the other, you go through a process of bringing the new one on and then, if that's the way you want, where you want two of them to operate continuously, you leave it there and, at some later time, you can remove one. And if you're simply try to change, then that intermediate time is reduced to the minimum necessary to do a stable and glitch-free transition.

Here's a rough picture of where we are. The square boxes indicate that things are in progress. Check marks indicate that things are done. You can see that a couple of things are done. And an awful lot is in progress. And a little bit remains in the future.

Here’s a picture of how a testbed would look like if that has multiple—in this case, two—independent signer-operators, one that we’re calling blue and the other that we’re calling green in this picture, each one of which as shown as having two publicly available nameservers that are serving the zone. And then you have multiple clients spread around the world that are watching and observing what’s going on.

So the components that are necessarily in a multi-singer controller are several. You have a finite state machine that is moving through the steps, keeping track of what step we’re on in terms of adding or removing a signer, an engine that’s driving all of that, and then APIs for managing the access and interacting with each of the components.

And here’s a similar scorecard for where we are. We’re tracking the development of proper interfaces on the DNS software packages, not—I’m sorry; PowerDNS, Bind, and so forth. And we’re also tracking nameserver software. It’s not PowerDNS, Bind, and so forth. [That] I think needs moved to [done]. I think that I’m actually a little behind in updating the slides. And then in the various DNS server provider capabilities, we’re also tracking progress there.

Here’s another version of the scorecard. This one shows that nameservers … This one is more up to date. It was updated a couple weeks ago. And there is room here, as you can see, for adding other nameserver software packages. So if anybody is working on this or has a progress report, let us know and we will make the point of adding it to this.

And similarly on the DNS server capabilities, things are in progress and moving along but not yet complete.

So there's a bunch of references. I've said this is the ninth we have of the agendas for each of the previous panels so that, if one wants to dig into this and see what the previous ones, we tried to make that as easy as possible.

So with that, thank you very much. And we will now move on to the next presentation. Brian, the floor is now yours.

BRIAN DICKSON: Thank you, Steve. I am now standing on the floor or whatever. So we're still undergoing beta testing on the DS polling, CDS, and CDNSKEY as an agent being the registrar of record for customers' zones that are being served by other DNS providers. And this is the overview.

Next slide. So this is a different version of the same diagram that Steve showed earlier. And it kind of calls out a couple different scenarios where there is existing methodologies available, where, for instance, we're the DNS provider. That's relatively straightforward. It's simply a question of either sending the data straight up to the registry, since we're both the registrar and the DNS provider in Scenario 1. There's no difficulties there.

In Scenario 2, that would be where the registry is doing the polling. And again, the publishing CDS/CDNSKEY works very well.

And so Scenario 3 is leveraging that general publication methodology and integrating polling of other third-party DNS providers for the same kind of information and using that to convert the retrieved records into the data that's sent over EPP. And that's what we're talking about in Scenario 3.

Next slide. And obviously this is just restating what those general steps are. Whenever there's a KSK rollover, the updates keys need to be converted into either DNSKEY or DS records and submitted up to the registry, depending on what kind of records the registry is looking for.

In Scenario 1, that's basically a vertical integration where we're publishing the records into the zone, but they're not actually being used from the zone. The records themselves are actually in parallel to that sent up to the registry through EPP.

Scenario 2 is where the registry is doing the polling and the publication is happening.

And Scenario 3 is where the third-party DNS provider does the same kind of thing, where, whenever the KSK is rolled, the CDS and/or CDNSKEY records are published into the child zone that's operated by that third party. And we poll for those records and, as appropriate, send the updates data to the registry.

And it's still not a lot of progress in the last three months. We're still in closed beta, but we'll just go over where we are and how the mechanisms look.

Next slide. Again, this is just a table highlighting who's doing what and what the requirements are. So this only works where GoDaddy is the registrar, but it works with any third-party DNS provider. And the third-party DNS provider does need to actually publish CDS and/or CDNSKEYs. And then the activity that's going on is GoDaddy's implementation, doing the polling of the child zones and submitting the DS or DNSKEY records to the registry for EPP.

Next slide. So we're still looking for additional participants. We haven't really been that active in the last month or two due to some competing resources and vacation and things like that. It's not a great time in the year for making a lot of progress, but over the next six months, we do expect to make significantly more advancements. And basically, we're just fleshing out the implementation. The basics are currently in place, but we're just making it more detailed and scalable to capture more state as things progress.

Next slide. The basic methodology is there's a lot of potential state about who the customers are, whether they're doing DNSSEC, whether they're using a third party as a DNS operator or GoDaddy. We need to filter those out. There's checking to see whether DNS records or CDNSKEY records exist, making sure the signatures are valid, checking to see if the CDS are new and haven't been seen before or, if they have been seen before, whether they've changed or not and then also checking to make sure that the CDS records are different from the existing DS records and submitting the differences through EPP to produce the new DS records at the parent.

The mechanisms are fairly deterministic. The actual control over the state is done. The control exists on the child side. So the child is expected to also observe when the parent changes have occurred in the registry and then follow through the stages of the CDS/CDNSKEY RFC to complete their set of state changes and migrate them from one set of DS records that correspond to DNSKEYs to a different set. And then our poller simply reflects the state of those keys in the CDS and CDNSKEY, and the loop is closed effectively through that process. So still we're not implementing that many of these stages, but we are in the process of doing more development on those.

I think that might be the last slide.

STEVE CROCKER: It was . Okay, thank you very much, Brian.

BRIAN DICKSON: Okay. Thank you.

STEVE CROCKER: Thank you.

Peter?

PETER THOMASSEN: So once again, hello. I've been asked to give an update on the automation of DS management, especially with regards to automation, to talk about all the statuses and recent developments.

Next slide. So I guess most people know that the DNSSEC validation rate globally is around 31%. Chances are, when you get a DNS response through your ISP's resolver, it will be validated by the resolver. And contrasting the secure delegation rate, the fraction of domains that are signed and also have DS records is only around 6%. Now, both numbers depend on the region where you're at. Some countries really push it to higher numbers. But what's problematic is that the number on the right is so low. Even if you look at DNS providers like deSEC, which we run and which is why I have the numbers, where we sign all the zones, the secure delegation rate is only less than 50%. So apparently there is something that keeps people from enabling it.

So let's look at the next slide and why so few delegations are secure. So deploying DS records is a multiparty problem. So it involves a bunch of parties, Necessarily the source of the key parameters, which is the DNSSEC signer. Usually that is the DNS operator also. And it involves the parent registry, which has to receive these parameters. But the communication of them is not direct. It is through the registrar and often also through the registrant. And the registrant however is usually not a technical person. They often don't even know about DNSSEC. And if they even hear about it and then try to get the DS records deployed, they're faced with a bunch of different kinds of web interfaces that are different per TLD or different per company when they have several domains with different companies. So it's very confusing. It's error prone. It's out of band and often not probably authenticated. So this is asking for automation.

If we do automation, we need to have the source of truth involved on one end of it. Otherwise, it's no use. So we have to make sure that DNS operator, which usually is the source for the key parameters, is part of the game.

Next slide. So this is an overview picture—slide 6—showing the flow of DS information during traditional DS deployment without much automation. So the first step is that the DNS service provider, who is usually also the signer, puts at the bottom right the signatures and DKSNEY information into their authoritative. Then the registrant fetches the parameters from the DNS service provider and relays it upwards to the registrar or sometimes, when they do have one, through the reseller. The registrar then continues via EPP to the registry who finally puts the DS records into the TLD servers. So the chain of trust is established. That's a lot of steps, and it involves people who are not part of the DNS hierarchy model, kind of. So the child operator is at the bottom, and the parent operator is at the top. And then there's the reseller and the registrant, who have kind of nothing to do with the DNS actually. And they're still part of the game and make it complicated. I mean, not to blame the registrant. They are not at fault for this. This is just how it has been set up. So a good way of automation would be if the middle layer could just be removed, and the top and bottom part is good to talk to each other.

Next slide. So this is an overview—slide 9—of methods for DNSSEC bootstrapping for securing a delegation for the first time and also for key rollovers. Both employ CDS or CDNSKEY records which have the C at the beginning because they love not in the parent zone but in the child zone, at the apex next to the SOA record, and they contain information about what the child wants the parent to publish in the DS records set. For the CDS records, that's just identically the same format. For CDNSKEY, when the parent processes that, they have to compute the hash for the DS record themselves. And the parent can discover this—for example, in a daily scan or whatever they chose to do. It needs to be consistent across nameservers to avoid harm. I talked about this earlier in the previous session.

And then we you have that, you're still lacking authentication, so for bootstrapping there is a proposal currently in the IETF DNSSEC Working Group where the CDS and CDNSKEY records are authenticated through an identically published copy of those records, which live under a subdomain of the nameserver host name which already in that scenario does have DNSSEC. So you can validate it essentially through the preexisting chain of trust to the DNS operator. So it take a little detour and you transfer the trust from the DNS operator. And this is kind of the only thing you can do because you don't have any trust into the child data yet. And the only thing that the parent knows otherwise about the delegation is DNS records. So that's all you can work with.

RFC8078, which did specify CDS and CDNSKEY for enabling DNSSEC for the first time did so in 2017 but in an insecure way—cryptographically insecure. There were extra precautions, like you have to look a few times from different vantage points and make sure the CDNS records look at the same. But that is not cryptographically secure.

If you do a key rollover, you don't need this complication because at that point in time you already have a chain of trust to the child. That's defined in RFC7344 from 2014. And as a parent, you can scan the child for these records and validate them as you would validate any other record from the child. And if you find there is updated content, you can roll the DS at the parent.

So those are the methods that are in place and cover all the use cases. And the one in the middle here is not yet finally specified as an RFC, but it's on the way.

Next slide. So when we have done this—these methods in place—then the middle layer is not any more involved in deploying DS records at the parent. So there's only three steps now. The service provider has to put the CDS records into the authoritative DNS server, then the parent does the scan (that is either the registry or the registrar)—I'm just showing one case here, but it's very similar—and then on the right-hand side, number three, the registry put the DS records in the TLD server or updates them there, which updates or established or updates the DNSSEC train of trust to the child.

Next slide. So the current state of deployment is … We should distinguish between the child side and the parent side. One second. So it is supported by DNS operators. Some support secure updating—that is … Actually, all of them support secure updating. So when you roll the keys, then they do validate the updates CDS records through the existing chain of trust. If you consider bootstrapping, some do the authenticated bootstrapping methods with the co-publication of records on other nameserver subdomains. Some don't do that. The ones that don't do that are DNSSimple and GoDaddy. At least I believe that still is the case. There probably are some others that I am not aware of, and if I am missing them here on the list, please let me know. Authenticated bootstrapping, on the other hand, has been implemented on the child zone by Cloudflare, which manages about 23% of the top million domains according to the Tranco list. So that's quite significant coverage. And deSEC—we're not aware of any other providers so far.

On the parent side, the ones that are doing the scanning—we have seven ccTLD registries that do it. Five of them do insecure bootstrapping. So they don't check the co-publication under the nameserver's subdomains. But two countries already do do that. They run [by Switch]—I mean, not the countries. Their registries are run by [Switch]. And it's .ch and .li. Chile is going to deploy that soon.

Also, GoDaddy is planning to perform CDS and CDNSKEY scanning as a registrar, as you just heard from Brian. And actually, as I found out earlier, there is another registrar who has been doing that since 2020, which is Glauca Digital. It seems to be a British company. So they announced it on their blog and otherwise have not appeared in any documentation about this. So I'll try to get that updated. I think there is some GitHub [tracker] that keeps track of this information. Yeah, it's actually linked here at the bottom and the source. I'll try to include that registrar there.

So this is the current state. And as you can see, there's a lot of room for other TLDs to adopt these technologies. .fo is going to do that in November, I think, on the 14th and then on other CentralNic domains (at least ccTLD domains) next year if I'm not mistaken. They have made an announcement about this.

Next slide. So it looks like the community could perhaps use some extra guidance on how to go about DS automation. And as a reminder, the point is to include the DNS operator who is the source of the DNSSEC parameters usually into the game so they can participate in the automation process. To tackle this problem, the Security and Stability Advisory Committee has established the DS Automation Work Party, who is going to work on the recommendation document that may or may not contain specific recommendations for how to DS automation. So it's not yet final. So far, the work party is investigating how things are currently done by registrars, registries, and DNS service providers, both manually and with automation. And then the plan is to explain the different methods and issues that exist with those methods and then perhaps provide specific recommendations on how to facilitate automation for DS record updating and also bootstrapping. The intended audience for this is essentially all parties concerned, which are DNS service providers and the registrars and registries. So that's the status of that. And we're working there to get that into a balanced recommendation.

I believe that was all. If you go to the next slide, that's probably that I would just take questions. But that's later, I guess.

STEVE CROCKER: Thank you very much.

PETER THOMASSEN: Thanks.

STEVE CROCKER: We'll move on to the next talk. And, yes, we'll take up questions at the end here. Alright, Roger Murray will now talk about the status of the multi-signer controller software, MUSIC, that they're developing. Roger, take it away.

ROGER MURRAY: Hello. Thank you. Thank you for taking the time to listen to our presentation. We couldn't get our expert in here today, so they sent me instead. I'm going to talk about MUSIC. It's a controller for coordinating and controlling the automation of a multi-signer process.

Next slide, please. We started work on this back in 2021 with help from DNS-OARC and [inaudible] in the beginning. And what we're trying to do is create a software that implements the multi-signer draft automation. That will be linked at the end of—well, its actually linked in your GitHub repo, so you can have a look there.

Next slide, please. MUSIC is working. It has been working for some time. It has two modes. One is a manual mode, which is basically using the CLI to push commands to the API and pushing the zones to the different steps in the process. And it also has an automatic mode that you can set signers and zones in that will just run everything through the system and keep an eye on the different states and move it along as long as everything is safe and secure to do so. But as always, when you're into the corners of DNS and trying out new stuff, you run into some issues.

Next slide, please. One of the issues we ran into was that the draft basically talks about the idea of just synchronizing the ZSKs, the Zone Signing Keys, and did not—and still does not; we need to update that—respect CSKs, which led us into a little corner that we to figure out a way to solve, which kind of led us to two alternatives.

Next slide, please. One alternative is to get out your pen and paper and do some heavy-duty engineering and figure out some really nice algorithms and very carefully and intelligently analyze the DNSKEYs to figure out what's being used for what purpose. Or you can do Alternate 2 and just synchronize all the DNSKEYs across all signers.

Next slide, please. We went with Alternate 2 for the following reasons. One, this is the simplest solution to the problem. It works. It also makes DNSViz very happy. Even with the smiley face, I want to give that some credit that, when it makes DNSViz happy, it also means that we're meeting the expectations of people that might be validating and things that you can't really see in the RFCs the way it's actually implemented in code. One of the downsides of this would be leading to larger DNSKEY RR sets, but we're basically leaning on the idea of people moving to ECDSA and then moving to key sizes down and that way. So the fact that we're synching a bunch of keys across a bunch of signers shouldn't be a problem.

Next slide, please. And this is basically information. You have to think about the size considerations, but as I mentioned, we're basically doing a punt on that right now and leaning towards just … It's going to be CDSA in the future for a lot of people or some type of small key size. That's what we're doing.

Next slide, please. Another corner of dust and crust that we ran into was that some signers are generating their own CDS records. So when we were putting up one CDS record, there was another CDS record already there. So what we decided to do is we're also including both SHA-256 and then SHA-384 when we update the signers. And this has really helped with consistency across the signers. It makes the code a little bit cleaner and just easier to keep an eye on everything. Johan and I and some other people in the working party are discussing ideas about what should the parents respect in regard to algorithms and who should be the deciding party there. So that may become something that comes up for more discussion later.

Next slide please, Steve. Part of the idea that we're going to have people testing the code … We realized that we need to look at how we were handling the database. We weren't using transactions, so we've implemented transactions so we can allow for simultaneous changes and more zones and signers being in the actual MUSIC controller at the same time. We still have a lot of testing to do here, but it's moving forward very well.

Next slide, please. Next steps is testing and bug fixes and then testing and then bug fixes and then rinse and repeat. We've had some really good progress in the last couple days, which is very promising. And as you hear, Steve is on us like the driver with the whip. And we've got a six-month window for the next time, and he's looking for some more significant progress. So hopefully we'll be able to meet that wish.

Next slide, please. Here's a link to where the code is, and then the read-me also includes links to the relevant RFCs and the draft for the DNSSEC automation. And if you have any questions, just drop us an e-mail or a chat. And then if you want to help out with any codes, we accept full requests/ideas. If you want to test, it runs on my laptop. And if you need any help, just get in touch with us. And thank you for your time. Thank you for listening.

STEVE CROCKER: Thank you very much, Roger.

Alright. Jan Vcelak on the activities within NS1.

JAN VCELAK: Thank you. So my name is Jan Vcelak. I'm a software engineer at NS1, and we recently added support for multi-signer DNSSEC. So this just a quick update of what we actually support as a managed DNS provider.

Next slide, please. So just real quick, RFC8901 describes two models for multi-signer. The main difference is that Model 1 uses a single key-signing key, which is managed either by one of the providers or externally. And then each signer, each provider, uses the zone-signing key. This was actually our first implementation.

We had this as a proof-of-concept about three years ago. Now this is discontinued because we think the real value is actually in Model 2, which provides more resiliency and is probably easier to operate. And having a single KSK probably doesn't improve security significantly because, in DNSSEC, the difference between key-signing keys and zone-signing keys is purely operational, but there is no technical enforcements on how the individual signing keys can be used. So for the sake of easier operation and resiliency, we would like to focus on Model 2. And this is actually what we added support for and what we are going to work on in the future.

So next slide, please. Just real quick, as a managing DNS provider, we provide traffic management features. Basically, it means that we tailor our responses to individual queries [inaudible]. And for this purpose, we have to sign the zones on the fly. So as the DNS server as responding to the individual queries, it's also generating a signature for the particular reply. [inaudible] like ECDSA. We use compact NSEC proofs. And the configuration in our platform is really minimal. It's either enabled or disabled. We don't let our customers, our users, manage their keys. We do it for them. So it's really just simple [chat] box in the portal

Go to the next slide, please. Here's actually how it looks. If you want to enable DNSSEC, just click on this [chat] box in the portal. Or we have management HTTP [API]. So the only thing you have to do is send this post. I [inaudible] zone RFC8901, the CZ, which I will show here in the examples, but this is the only thing you need to do to enable DNSSEC. And [inaudible] automatically bootstrap the keys and immediately will start seeing some responses from our DNS servers.

Next slide, please. So the supporting multi-signer Model 2. The only thing that is needed is actually the ability to allow adding additional DNSKEY records into the zone because, as I said, NS1 is managing the DNSSEC-signing keys. So we are also generating the DNSKEY records for the zone. To allow multi-signer configuration, we have basically added the new end point into our API, which is here below. There's a link to the [inaudible]. I don't want to go into too much detail. But basically by using this end point, you can create, update, delete additional DNSKEY records that should be published alongside the other DNSKEY records, the ones that are managed by NS1.

Next slide. So here's actually an example. This domain is actually live if you want to play with it. Please have a look. I use it for some testing. So maybe it'll be slightly out of … The configuration is different as it's presented now, but at this point, it's exactly what I'm presenting. So I have this testing configuration using NS1-managed DNS and an open source of DNS server. I've configured Knot DNS for signing the zone, and I imported the NS1 keys, like public keys. And this is what you have to do on the NS1 side. These are the keys that Knot DNS uses for signing. The first one is the key-signing key. The second one is the zone-signing key. So if you send just this API call, it will set up everything for multi-signer on the NS1 side.

Next slide, please? Here's just an overview of how the zone is configured. The delegation is set up to point at my virtual server, which is running Knot DNS and NS1. And if you query for DNSKEY records, it doesn't matter which server you query. You should always get the same answer, which is the trick behind multi-signer DNSSEC. So you will see that there are two keys for NS1, two keys for Knot DNS. And the zone was set up almost identically. Like there's no zone transfers configured or anything. It's like really independent in configuration. So you can see that if you query for a txt record, you will see what the response are generated by Knot DNS or NS1. And if you look closely at the signature, you will also see a different key pack, as you will see that the answers from Knot DNS are really signed by the zone-signing key of Knot DNS and vice-versa with NS1.

Next slide. Here we have just the same configuration, [cz] and DNSViz. You can see that DNS DNSViz is really happy. It's not showing any warnings or errors. There are two DS records, one DS records for the KSK for NS1, the other one for the KSK I set up with Knot DNS. You can see the zone-signing keys. And all the answers are signed. There's a permanent link at the bottom of the slide if you want to drill down into the responses from the servers and just investigate.

And on the next slide, I think I have some closing remarks. Yeah. So from the NS1 perspective, I think this is what we imagine for multi-signer support. It basically works. At the moment, directional keys can be managed only by the Rest API. We don't have the support in the management portal, which we will probably add eventually. But if you're looking for automation, Rest API is what you need. We don't support CDS and CDNSKEY now, but we are working on it. There's no technical obstacle for this. It's mostly a product decision. As I mentioned initially, our configuration of DNSSEC is really simple. It's either on or off. And we are just trying to decide or we want to decide whether it's safe to just publish CDS and CDNSKEY records for all zones or if we have to provide a configuration option for [NS1] as well.

One missing piece—it's not really a missing piece—is (we just heard a presentation about MUSIC) is orchestrating the updates on the keys. This is of course missing, and we are thinking about having something like MUSIC within out platform, which would allow integrating with [inaudible] providers. So we would like some kind of process to manage the keys between the providers on behalf of the customers, but we don't have any timeline for that. But we are considering it. And, yeah, we have validated implementation with BIND, DNS, deSEC. And Cloudflare has it as well, although it's future flagged.

And if you're an implementer and if you are thinking whether you should start, I think it's much easier to focus on Model 2 because the interface is simpler. The only thing to support Model 2 is really the ability to add additional DNSKEY records into the zone. And you don't have to modify the singing process for the zone in any way. For Model 1, I think it's more complicated because you also have to modify the signer because sometimes you don't want to sign DNSKEY records and you just want to accept [inaudible] signatures generated by some third party or some other software. But that's my personal opinion.

And if you are looking for implementation, just feel free to reach out. I'd be happy to discuss this, especially if you want to integrate with NS1 in any way. That's it. Thank you, Steve.

STEVE CROCKER: Thank you very much.

Alright. I'm going to wrap up with a few thoughts about current plans and on testbeds and scenarios. A useful exercise would be to set up testbeds and have the scenarios of adding and removing the zones, adding and removing signers, on a continuous basis so that the test get run over and over again.

What’s the value of that? The value of that would be at least two different things. One is it would make it possible for others anywhere around the world to look and see these operations in progress sort of like a museum display that you can just come along and look at any time and it’s operating. The other is that it would provide some data about how long these transitions take and whether or not there is a high degree of similarity or various lengthening of the transition times, which would then lead to some questions about, why is that?

So that’s kind of the goal. We haven’t quite gotten there yet, but that’s a thought about how to proceed. We’re kind of in the early days.

So the multi-signer has a number of processes and steps. And the primary processes that we’re talking about is adding a DNS signer and removing a signer. And then there’s a series of things that are needed to implement each of those.

We’re at a stage—and I’m speaking on behalf of Roger and Johan—where the testing is taking place within their institute but [is] for others to mount the software and try it out. So this is a definite ask here. If anybody is interested in participating, please speak up, and you will get quite a bit of attention.

So as I said, a future scenario so that we’ll have sequenced transitions, continuous repetition, observations of hopefully successes (but it might also be possible to observe glitches, which would indicate something is going wrong, either a software failure or a configure failure or something else that we would learn from) … When I prepared this slide, I was focused on the idea that these transitions take place on a timed basis. They actually take place on an event-driven basis. So it's not timing driven. And finding timing errors hopefully can't occur. On the other hand, on an event-based, what happens is the amount of time it takes might vary from one instance to another. And as I said, one might learn something from all of that.

There are a number of open issues for how to operate all of this in production mode. What happens if you need to do a key rollover in the middle of a transition of adding or removing a signer not currently supported? There has to be CDS/CDSNKEY scanning. And the parent needs automated operation of all of this, including CSYNC for production operation.

So this is just a summary of what happens if you change the DNSSEC keys in a zone. A normal key rollover constitutes a series of changes. A multi-signer process constitutes a different set of changes. And if you're trying to do them both, then they need to be integrated in a careful way, and that has not yet been carefully explored.

Here's pictures that suggest all of this. There has been a parent zone assigned for test purposes, multisigner.se. And then there will be multiple delegations underneath that for test purposes. And the TTLs on these will be shortened from what they normally are at a top level so that everything can proceed at a much faster pace and you can observe how all of this works.

Here's the same picture about test beds, which you've seen before. And currently all of the testing is being done by [Johan] and Roger in their laboratory. And that brings back the question of, how do we expand the test basis? And so, again, if anybody is interested, please speak up.

Here's the status slides that I showed before at the beginning. And these are being updated as we get new information. And here's the components. So all of this is, for reference, just the repeat of what we had before.

So that's the summary of the testing and scenario generation process that we're looking for. And as has been mentioned more than once during this panel, what we're looking for for the next time with this six-month time between now and the next panel is some interesting and hopefully very positive results on both the implementation and the deployment and the testing process.

So thank you very much. And we have … I think we're over time. I don't know, Kathy, where we stand with respect to any time for questions. And I don't have a sense of whether people want to ask questions. So let me turn it over to you.

KATHY SCHNITT: You can go over for a little bit, Steve. It's fine.

STEVE CROCKER: Good. And is there anybody wants to ask a question? Or do any of the panelists want to add any comments?

As I've learned in hanging around diplomats, I hereby declare this meeting a success. Thank you, all. We'll see you in six months.

[END OF TRANSCRIPTION]