ICANN85 Mumbai | CF – DNSSEC and Security Workshop (2 of 3)
Wednesday, March 11, 2026 – 15:00 to 16:00 IST

KATHY SCHNITT

Thank you, Kahlil. Hello, and welcome to the DNSSEC and Security Workshop: Part two of three. My name is Kathy, and I am joined by my colleague Dan, and we are the participation managers for this session. Please be advised that this session is being recorded and is governed by the ICANN Community Participant Code of Conduct, the ICANN Expected Standards of Behavior, and the ICANN Community Anti-Harassment Policy. To help keep this discussion moving smoothly, here is how you can get involved today. To keep the queue fair for everyone, all participants, including those here in the room, should use the raise hand feature in Zoom. When speaking, please state your name and affiliation for the record and maintain a moderate pace. You may also submit any questions for the speakers today through the Zoom Q&A pod. We will read those aloud as time permits when directed by the moderator. While the chat is available for attendee interaction, please note that comments or questions posted in the chat will not be read aloud. Official questions must be raised through the speaking queue or submitted through the Q&A pod. And with that, I am happy to hand the floor over to Peter Thomassen.

PETER THOMASSEN

Hello, this is Peter Thomassen. I will be moderating this panel. I will be quickly giving you an overview of what is up, and then we will dive right in.

Very briefly, what is DNSSEC automation? There are multiple aspects. One is signing the zone and generating keys. Most DNSSEC signing software has had tooling for that for quite some time. The other aspects are automating the provisioning of the DS record, which is a more complicated problem, especially when the DNS service is not provided by the registrar but by somebody else. This panel is partly about that. Another part is about so-called multi-signer setups, where you have two signers that are independent, and they serve the same zone.

A quick update on specifications. There are actually several specifications in the IETF that are being worked on that relate to DNSSEC automation. One is about CDS and CDNS key record consistency checks that has to do with ensuring that update requests are not ambiguous. Then there is a draft that recently finished the working group last call in the DNSOP group at IETF about operational recommendations for certain implementation details of DS automation. There is a new draft recently adopted for how to do DNS delegation management in general using the DNS update message format, which is not based on CDS records but uses another approach. That is quite new work, so let's see how that develops.

Here is the panel that will be speaking today. It looks like I have messed up the format. Anyway, there will first be a presentation on the multi-signer transition of postfix.org and related domains from one provider to the other without any downtime. Then there will be a presentation about generalized DNS notification implementation in BIND, a report on the implementation updates from Switch, and a presentation by Barbara on how stories and emotions shape the adoption of technical standards. I will give it over to Carsten and Patrick to start with the Postfix presentation. Feel free to share your screen.

CARSTEN STROTMANN Hello. Good afternoon or good morning, wherever you are. Is the audio clear and good?

PETER THOMASSEN It is great.

CARSTEN STROTMANN Okay. I would like to tell you a little bit about our experience while moving the Postfix domains. We had to move some DNSSEC signed domains between signers, that is, different organizations that do the DNSSEC signing. These domains are for the Postfix open source mail servers, and Postfix was the first mail transfer agent that supported DANE. People that are interested in security, interested in DNSSEC, and interested in email security are watching, so what we do there is kind of a role model. There are always people who

closely watch us failing and then claiming that if they screw up DNSSEC, nobody can operate DNSSEC. There was a little bit of pressure.

Who are we? There is Patrick, who cannot be here today because of private reasons. He wrote a book on Postfix. He is in the email community for a long time, and he is an active proponent of DNSSEC and DANE. Then there is Wietse Venema. He is the primary author of the Postfix software, and he is the owner of the domains. Then there is Peter Thomassen, who is the co-operator of deSEC, and that is the DNS hosting service that we are moving towards. And it is me, I am the admin that oversaw the work and everything we did there for the Postfix domains.

Originally, all the Postfix domains were hosted on two DNS servers operated by Wietse himself. That started around 1998 when the Postfix project started and the first release was made. Then in 2022, for some reason, distributed denial of service attacks against the DNS infrastructure of postfix.org, which is the main domain for Postfix, happened, and that resulted in the website and the mailing list and the mail service itself not being reachable for some amount of time. We do not know why postfix.org as an open source project has been selected for denial of service attacks. We can only guess that maybe because of the high visibility, that was like a testing ground for the DDoS operators to see if their system really works or maybe to show off.

When the denial of service attacks happened, we offered to help with our DNS expertise and also with more DNS infrastructure to fight the denial of service attacks. We did that by moving the domains from the two DNS servers operated by Wietse in the US to a DNS service that we spread around the world. The attacks continued in 2023, but after the move to the new infrastructure, they could not bring down the domains anymore. One reason for that at the time was that the denial of service attack was mostly IPv4, and what we did is we deployed some of the authoritative DNS servers for the domains IPv6 only, and that made them unreachable for the attacks. However, the domains were still visible to most users because most resolvers out there, especially the big ones, are dual stack, so they can reach IPv6 only hosted authoritative servers. Then the denial of service attack stopped in summer 2023 and never came back, which is a relief.

In the time between 2022 and 2025, we had six DNS servers, two in Europe, two in the Americas, and two in Asia. They were traditional networking. There was no Anycast, nothing really fancy there. We signed the Postfix domains in 2023, and the DNS servers were running on either Ubuntu LTS with BIND 9 or OpenBSD with NSD. Now, because of the NIS2 regulation in Europe, Sys4 as a company cannot afford to offer pro bono DNS hosting anymore. The reason is that because of the NIS2 regulations, everyone operating a DNS hosting for someone else needs to have certain reporting requirements and security requirements. We as a little company cannot really do that. The decision was made to move the domains

to deSEC, which kindly offered free and possibly much better DNS hosting because of Anycast and all the better expertise than we have. We are happy to take the offer and move to deSEC with the Postfix domains.

However, we wanted to move without losing DNSSEC at any point of time because, as I mentioned before, people are watching. They are highly visible. The domains are highly visible in the community of a lot of mail and DNS admins. We do not want to remove DNSSEC during the transition. Why is this a challenge? deSEC has its own DNSSEC signing infrastructure, and there is no easy way to import existing private keys to reassign the zones with the same keys. We had to do a key rollover while moving.

What we did is we imported the zone data minus the keys and the signatures into the deSEC system. Then we imported the current key signing key public key, the DNS key resource record into the new zone, and we imported the new common signing key from deSEC into the old zone on the hidden primary on the old infrastructure. We waited for both DNS key sets to be updated in the resolver cache. We waited the time to live of the DNS key record set plus some buffer time. Then we added the new DS record in the parents, and we waited for that DS record, the old and the new, to be in the resolver caches. We waited the time to live of the DS record set, and then we switched the NS record set for delegation to the new infrastructure at deSEC. We observed and did

monitoring, and when everything was good, we removed the old DS records from the parent and removed the old infrastructure.

The important part comes now, what we have seen during testing. Postfix.org is the main production domain, and there is postfix.com, .net, .co.uk, and .org.uk. These are domains that are reserved, but they are not really used in production. There is no mail server and no real website other than a CNAME pointing to the main website. We had these four domains to prepare and test the migration and to see if everything was working smoothly before we attached the highly visible postfix.org domain. None of the parents offer DS record automation. Everything needs to be done manually. Neither .org, .net, .com, nor the UK domains offer DS record automation at the point. What made stuff a little bit harder is that we had different registrars, so different web UIs for managing the DNSSEC DS records for updating them in the parent.

This is one issue that we encountered while testing with postfix.com. What you see here is the old DS record there, and the web interface had a button to add a DNSSEC record, which should mean add a new DS record. If you read "add", you might expect that this will add a new one in addition to the existing one. But what really happens if you use this button is it will replace the old one with the new one, and by doing that, it will immediately destroy the DNSSEC chain of trust. We found this out the hard way, and postfix.com got insecure for about five minutes. We had monitoring in place, we were immediately aware that something was wrong, and then we switched back to the previous DS record, restoring the

chain of trust. That worked for five minutes, and of course for the TTL of the DS record for all the people who fetched the DS record during those five minutes, they were in pain. Unfortunately, postfix.com is not an operational domain, so I hope there was not much harm done there and it was not really visible. We had to have a conversation with the technical support of that registrar and, after lots of back and forth explaining what we want to do and why, they were able to add the extra DS record manually into the parent zone. Of course, that is nothing that a normal customer would do, and also that does not scale.

Another problem that we encountered was that with a different registrar, managing the DS and NS records was a challenge because the web UI would show a different state than what was transferred over to the parent. We would add all the DS and NS records that we needed during the transition phase, and in the web UI it all looks good. Then you hit submit, and then you observe the state of the changes in the internet in the top level domains, and it is different from what you have entered. It was just hit and miss. Sometimes we had three out of four NS records, or we had one of the two DS records, but not always the same one missing. It was either the new one or the old one. It took five iterations, and by chance, at some point of time, we hit the correct setup. It was really painful and of course, that also destroyed DNSSEC for that domain for a certain amount of time. Fortunately, we did not have to use that registrar interface for the .org domain, for the important domain. But if people say DNSSEC is complicated and error-prone,

these kind of bad user interfaces do not really help. They contribute to the narrative that DNSSEC is hard to manage and that people should not touch it because it is too dangerous and it will destroy the DNS connectivity.

Conclusion here is that moving a DNSSEC signed zone is a challenge, mostly because of the bad management of registrars. DS record automation would really help. I have some domains in Switzerland and one in the Czech Republic. They offer DS record automation. I have done key signing rollovers and migration there, and it was a breeze. It is really nice. Unfortunately, the big top-level domains do not have that at the moment. Another takeaway is that registrars unfortunately also treat DNSSEC as an exotic add-on that nobody really wants, so they do not really invest in good web user interfaces and training for their support staff. That is my report, and I am up for questions.

PETER THOMASSEN: Thank you very much, Carsten. We can take an urgent question if someone has one, and the bulk of it I would like to move to the end of the panel. Go ahead, Weiss.

WEISS: Thanks. Fascinating presentation, and having gone through that exercise, it is non-trivial. I actually wrote an internet draft for the IETF at one point that talked about the proper way to potentially go insecure for a little while if you are in a zone that may not have

the expertise that you just went through. I would love to hear your thoughts on what is the required expertise level that you guys had to know as opposed to sometimes it might be easier to mitigate it by setting TTLs really low and being offline for five minutes, whether it was accidental or not. It is interesting to note that you even made a five-minute mistake and you did not intend to. Any thoughts?

CARSTEN STROTMANN Yes. Certainly for a majority of domains, going insecure is an option. It was not an option for us here because of the visibility, and also we wanted to prove that it is possible. It really depends on the security requirements of the domain and also whether DNSSEC is being used as an enabler for some other security protocols that might be attached to the domain. Depending on that, you might not be able to go insecure, because if you go insecure, that might mean that other security services are then vulnerable or do not work anymore. But I would agree that for a large majority of domains, going insecure for a short amount of time is a valid option.

PETER THOMASSEN Thank you very much. If there are any other questions, let us have them at the end if there is time or offline. With this, I believe the next presentation is by Matthijs Mekking. Have fun, Matthijs. You have 15 minutes as well.

MATTHIJS MEKKING Hello. Am I audible?

PETER THOMASSEN Yes.

MATTHIJS MEKKING Great. I am going to talk about generalized DNS notifications in BIND. I made an implementation there, so we have a new version coming up this year with new stuff, and one of them is generalized DNS notifications. The gist of DNS notifications is similar to the one we already know, which is the notify SOA, where the primary signals to the secondary, "I have new updates in my zone. Please do a zone transfer." This RFC nudges a parent basically that the child, the delegated zone, has new information that needs to be updated in the parent zone. Notify CDS is a type of notify to tell there was a key rollover probably, and then the delegation signer record needs to be updated to match the new key. The notify CSync is a way to tell the parent that the NS records and the glue has changed. The RFC comes with a new DSync record, and that is used to discover notification endpoints, so you do not have to configure them in your named.conf for BIND.

There is a record, and I have two examples here. The first one would be a way to find the notification endpoint in a tld.example, and it is eligible for the delegated zone test.example, and then it can send its CDS notifications to cds-scan.example listening on port 5359.

The second one is in a wildcard, so that would be a way to have a notification endpoint for all your delegated zones.

In BIND, how is this implemented? I am going to quickly go over this because I have little time and I have a demo prepared. Suffice to say, there was code refactoring going on, moving all the notify stuff into this one place, creating a context out of it, so we can now duplicate a lot of code to introduce new notifies. That is what I did for notify CDS. The same thing applies for zone fetches because, to find the notification endpoint, we need to fetch some zone data. We need to find out where the zone cut is, which is an NS fetch. Once we found the name server of the parent or the name at least, we can do a DSync fetch, which you see constructed with child prefix, DSync, and parent suffix.

Looking at the existing notify configuration in BIND, there are a couple of options related to notify. Obviously before this change, everything was applicable to notify SOA only. I have commented out some of the options here because they need special treatment. We will show that in a bit because I am going to show now what has changed in the development release. Looking at the right side block, we have introduced a new notify configuration clause. Rather than adding notify-cds-defer and notify-cds-delay, we thought that was obviously not the way to go, so let's introduce a notify configuration clause where you can parameterize it with a type and then set all these options in there. Looking at the notify config SOA, you see a lot of similarities between the old and the new ones, and you can still use the old ones. But if you configure a

notify configuration clause, it will overwrite the old options. Now you can also configure a notify configuration for CDS, but not all those options are translated one-to-one. For example, a notify for zone that is explicit and primary only, as you might have guessed, are values that are applicable to zone transfers only. For CDS, we can only allow yes and no, so it is a Boolean value. Also, things like defer and delay are in the configuration if you have high updates in your zone and you want to manage the rate and how many notifies you are going to send out within a certain duration. With CDS, it seems less important because key rollovers are done once a year, or at least the KSK rollover, so not very often. That is the thought. We are going to ignore those values for now. But we are going to keep them in this notify configuration clause because these are values that may be used in the future for other generalized notifications.

A word on the other notifications like allow notify. I implemented BIND in sending generalized notifications, but not receiving. Allowing notify for CDS does not really have any value. Similarly, also notify is a way to notify other endpoints, but it is not required for CDS because there is a mechanism called DSync, and the DSync records provide the endpoints where to send the notifies to. The options on rates are global options, so they are not specific to a typical zone. They are global options, so they do not need to go into the notify configuration, and the value that you set there is applicable to all queued notifies.

Summary, we have a new notify configuration option to configure generalized notifications. The one-to-one mapping is not really a one-to-one mapping. There are some corner cases. Rate is a global option, and we do not allow receiving notifies. Why do we not have receiving notifies for CDS? We expect a dedicated server to listen to those notifies. If you feel that this function should be in the generic name server and if that is going to be community consensus, we are open to adding that feature. I have been talking a lot about notify CDS because we have not implemented notify CSync yet. Sending would be trivial to add, but when to send it, the trigger is very different. BIND implements automated DNS signing, so we inject the CDS and CDNS key records during key rollover, and those are perfect times to send the notify for the CDS. But for CSync, this is done by the operator and, in order to trigger a notify CSync, that would need to have some hooks on RRsets when your zone loads. That is a major refactoring and a whole different project. I think it will be happening because we want that for multi-signer support as well, but it is a whole different project.

I do not know why I did this to myself, but I prepared a demo, so I hope it will be going all right. The demo basically shows a local setup where I set up an example TLD, a delegated test.example zone, which is automatically DNSSEC signed. It publishes the CDS and CDNS key records and sends the notifies. It also checks the parent for whether the DS has been published. We have a dedicated scanner which listens to notify messages and then queries for the CDS records in the child zone. Once it finds those, it updates the

delegation signer with the DNS update. Then we have the parent zone.

What we are going to see in the demo is basically the start of a key rollover and all those steps that I just explained. Main thing is that it will be all without manual intervention. You probably see now a couple of terminal screens.

PETER THOMASSEN: Matthijs, you have four and a half minutes left.

MATTHIJS MEKKING: Okay. I have a name server for the test.example zone. Since we have a couple of minutes left, I am not going to show the configuration. I am just going to start the parent server, start the test.example, and what I show here is it introduces the key. This Python script is the scanner. It is listening on this port, which matches the DSync record. Up here, we are going to watch the CDS record and the DS record at the parent. The demo takes about four minutes, so that is fine. What we see here in the status is that the zone is going to be introduced. Currently, it is waiting for the zone to be fully signed before submitting the DS to the parent. That is in about ten seconds. What you will then see in this Python script is it will receive the notify CDS. In this bigger window, there are the logs of the named. It will send the notify to the CDS scanner. It receives here, and this scanner will now dynamically update the parent. Over here you can see this is the CDS in the test.example zone. The

dynamic update is being done, so in a while we will see that the DS will be published. This dig script will show that it is over here. Now the test.example will do the rollover, checking that DS is enabled, so BIND will check the DS at the parent. It saw the DS. It is now waiting for the DS TTL, and then the rollover is complete. The demo was very fast. I hope it was easy to follow despite the speed. I am going to let it run for a couple of minutes because in a couple of minutes it will do a rollover, but in the meantime, I am happy to take questions.

PETER THOMASSEN Very cool. Last time I did a demo it took a few hours because of publication and scanning times. We have about one and a half minutes for questions, if there are any. I do not see any in the room or on Zoom. I guess we just win a minute, and we believe you that the rollover works. Thank you for doing this, and it seems like a very nice update to me. Thank you, Matthijs. All right, next up is Oli Schacher with an update on the DS automation implementation by Switch, which operates .ch and .li.

OLI SCHACHER Yes. Let me see if I manage to share stuff. Bad news. My Mac is complaining that I need to restart Zoom. Hang on.

PETER THOMASSEN It is not a problem. We can share the slides for you and advance them on your request.

OLI SCHACHER I should be fine with a quick quit and reopen. Hang on one second.

PETER THOMASSEN Are you still there, Oli?

OLI SCHACHER Okay. If all is well now it should work, hopefully.

PETER THOMASSEN Okay, you have eight minutes.

OLI SCHACHER Perfect. Great. I would like to talk a bit about our DS automation in .ch and .li. When you start implementing DNSSEC automation, there are quite a few decisions that you have to make on how to implement this. Currently, as Peter already mentioned, there is a draft that tries to give recommendations on these decisions. For example, should you scan CDS records or CDNS key records? When we implemented this automation in 2019, there was no such document, so we had to figure this out all by ourselves. Luckily, we were only the second TLD to implement this. The first was actually .cz, and they had the first-mover disadvantage. They could already tell us a lot about their decisions and how it impacted their operations. We could prevent a few pitfalls. With that, I would like to go over those recommendations currently in the draft and compare them to our current implementation just so you can see

how this aligns and where we actually made changes or where we decided things otherwise and why.

First of all, as I already mentioned, there is the question of whether you should scan CDS and CDNS key. The draft does not actually make a recommendation on that, but it says you should check them both and compare them for consistency. It says when you publish both CDS records and CDNS key records, the parent should make sure that they basically tell you the same thing. We are not currently doing that. In this whole automation thing, this is one of the big problems, that there are actually two record types that say the same thing. It causes a lot of confusion. What does it mean if you publish only CDS records? What does it mean if you publish only CDNS key records? What does it mean if you publish both? This is really undefined, and the question is, as a child zone, do you have an impact on what the final DS RRset actually looks like? Can you decide what hash algorithms, for example, will be published? This depends really on all of this. There is a lot to be still defined. What we decided is we just scan CDS records, and we accept the CDS RRset and publish it as is if we think it is okay, or we reject it. We do not look at CDNS key at all. We do not compare the two. It would mean we would need to scan a lot more records. We have to scan two and a half million domains each day. Adding CDNS key records to those scans would increase the scan time. Currently, we are not doing that.

PETER THOMASSEN You have four minutes.

OLI SCHACHER I will make it quick. Name server consistency, check all authoritative name servers of the child zone instead of just one. We are doing that since the RFC was published regarding that. This is a change that we made based on new RFC developments. Continued DNSSEC validation, the draft recommends that you should check as a parent whether the domain still validates after you made the change in the DS RRset. We are doing this. This is called the DNS resilience program. We scan every .ch and .li domain each day, and if there is a DNSSEC validation failure for any reason, then we notify the registrars and tell them. Because this has a financial impact, since they get cheaper domains if the domains are signed correctly, they actually read those emails and fix broken things. Also, one recommendation is that you should reduce the DS TTL after you made a change. This allows for faster recovery time if anything breaks. This is our newest change; we implemented this at the end of last year. Now, if there is a change to the DS RRset, either by the automation or manually through EPP, then the DS TTL gets shortened to fifteen minutes for the next twelve hours.

Reporting and transparency, it says you should inform the people when you make changes to the DS RRset as a registry. We are not actively informing them. Instead, we have a web portal where they can check for updates. This has to do with recommendations from

.cz. They told us if you actively inform domain holders, they usually do not know what you are talking about.

PETER THOMASSEN Two minutes only.

OLI SCHACHER I will make it very quick. Report queries we do not support yet. Change poll notifications, we do support them. We tell a registrar if we made an automated change from the registry. They can see the active DS configuration. The draft also recommends that you should see the DS history. We do not have that currently, but the .ch zone is public, so there are a few services that provide this for us. This one is a short one: should you make automated changes if there is a registry lock in place? We also decided that a registry lock is basically there to prevent a domain from being stolen, and we think you should still be able to make automated changes to the DS RRset. Finally, the last chapter is that you should provide a manual update channel for DS maintenance. That is crucial because automated DS management only works as long as the zone is actually correctly signed and you can read the records. We have that through EPP, obviously, so every registrar can always make changes. This update should be executed immediately. Unfortunately, we cannot do that since we have a scanning batch job that runs once a day, so there can be quite the time between the scanning and the actual updates. That is basically a design mistake that we made and that I would like to change at some

point. The last thing is that you should suspend DS automation if there is a manual update. We are doing this when a domain is currently unsigned for automatic DS updates. We have a three-day timer where we check the domain if there is a change. When you make a manual change, those timers get suspended and basically restarted. We implement this in a way. I think my time is already over.

PETER THOMASSEN That is correct.

OLI SCHACHER But-

PETER THOMASSEN But it was a very informative update. Thank you very much, Oli. We do not have time to take a question, but the chat is available. Maybe if someone has a question, you can get a response from Oli there.

OLI SCHACHER Sure.

PETER THOMASSEN I will hand it over to Barbara, who will be talking on another topic around this topic.

BARBARA JANTZEN Will you share the presentation?

PETER THOMASSEN Yes.

BARBARA JANTZEN Hello. My name is Barbara Jantzen, and I am a colleague to Peter Thomassen at deSEC in a project focused on DNSSEC automation, which is supported by the ICANN Grant Program. Taking a step aside, how stories shape the adoption of technical standards, or in other words, how image can get in the way of factual appreciation, that is my topic today. You are very lucky because last time I gave a similar talk at DNS-OARC, I started out by tricking the audience with an impersonation. While introducing myself, the project I am working on, and the topic of my talk, I spoke in the disturbingly shallow way of bad marketing people. The audience uneasily shifted in their seats, marveling at what improbable UFO had seized the stage until I confronted them with what was going on. Although everything I said was true, my posture had made them discard what I was saying as not worth their while, thereby revealing that hard facts alone could not account for their assessment of my statement, and that they were subject to solid prejudice, just like everybody else in this world. But I will not repeat the act, so you guys are lucky. But you are also unlucky because that means that you will not experience the trap that you would

have gone for and gain the quick recognition that you are definitely in the scope of what I have to say, and instead, you will have to believe me, and I know that is not your strong suit. But believe you must because these are the insights that you are looking for. If now you are angry because I deprived you of that experience or because of what I just said about your reluctance to believe, or simply because you do not like the way this is going, keep that anger alive because emotions are at the heart of what we are going to talk about.

This talk is about why image and stories are worth keeping an eye on, and it is based on the example of DNSSEC. I will state common perceptions of DNSSEC, give a broad outline of the history that led to these perceptions, sketch both the challenges DNSSEC supporters are facing today and the potential approaches to addressing them, and finally, provide several lines of inquiry that might benefit ongoing and future development and deployment endeavors such as DELEG, because DELEG has a certain aura. It is mildly reminding of Beckett's Waiting for Godot: "He should be here. He didn't say for sure he'd come. And if he doesn't come, we'll come back tomorrow." So much for DELEG for now. Let's dive into DNSSEC's narrative situationship. Jumping right into it, what is the image of DNSSEC? It is not pretty, and all this will probably sound familiar to you. I will just browse through the list without much comment. It is old, it is uncool, it is dead, it is complicated to implement and to use, expensive, it causes outages, it burdens the camel. We will get back to that later. It is unnecessary. It does not

do confidentiality. Of course it does not. It is not perfect. It is not what our customers want. We heard it. It is unheard of sometimes. "DNSSEC what?" That is the best of all scenarios. Nobody uses it anyway, and it cannot be good because if it was, it would be much better deployed. That is what you call a vicious circle. The point where all hopes for salvation are buried.

How did that come about? Let's have a brief look at DNSSEC history. As we are all worshipers of the goddesses of both rationality and causality, let's tell this story in the "of course" mode, meaning this one thing happened, and then of course, that other thing happened. You will see it will mainly work, and when it will not, it will get interesting. Of course, you will help me because after all, this is a workshop with built-in participant activity. When I make this gesture, you will have to look at me and say, "Of course." Okay? Let's give it a dry run.

AUDIENCE

Of course.

BARBARA JANTZEN

Of course. This was not so dry. This was very good. Let's jump into it. Back in 2005, with security awareness rising, the RFCs 4033 and 4035 defined the DNSSEC protocol designed to guarantee the integrity of DNS answers. Because of that...

AUDIENCE	Of course.
BARBARA JANTZEN	It is regarded as a useful protocol likely to be universally deployed. This usefulness...
AUDIENCE	Of course.
BARBARA JANTZEN	Leads some people to promote DNSSEC in a rather patronizing and polarizing way, like, "If you don't do DNSSEC, you're not doing security right." In turn, still other influential people...
AUDIENCE	Of course.
BARBARA JANTZEN	Got wound up about DNSSEC's necessity and peddled it to everyone, regardless of their technological maturity, leading...
AUDIENCE	Of course.
BARBARA JANTZEN	To massive outages. That is right. As the tech community is very fact and number-driven...

AUDIENCE	Of course.
BARBARA JANTZEN	These outages got listed in detail, namely on the IANIX website.
AUDIENCE	Of course.
BARBARA JANTZEN	This led to a massive drop in trust, resulting...
AUDIENCE	Of course.
BARBARA JANTZEN	In low deployment rates for DNSSEC. Of course, these outages also drew some other people's attention to DNSSEC's actual or alleged failures, who in turn felt that they should pinpoint DNSSEC as enemy number one. And...
AUDIENCE	Of course.

BARBARA JANTZEN This set a DNSSEC bashing trend. Of course, very effectively relayed by internet longevity. Given the obvious difficulties users of DNSSEC were facing...

AUDIENCE Of course.

BARBARA JANTZEN Experts have continued their work on DNSSEC to allow for a smoother implementation and maintenance, namely via automation. And...

AUDIENCE Of course.

BARBARA JANTZEN Nobody noticed. With the result that, of course, in 2026, secure delegations are still scarce. There was a little logical bump there, wasn't there? Let's get back to it. In response to the massive availability problems caused by flawed implementations and maintenance of DNSSEC, DNSSEC people continued their work on the protocol to allow for smooth and reliable implementation and maintenance, namely by automation. But not only did people not reconsider their stand on DNSSEC based on the new developments at hand, they did not even notice them. That means they got impacted by image, by the story that was being told.

Let's grab this idea of story and go back to our DNSSEC history. I am going to make it quick. We will see that story is everywhere, and the most interesting item this time comes first. DNSSEC started out as a no-brainer, so people were naturally inclined to think that it needed no story. Even more so, it did not even occur to them, and that seems quite sensible, does it not? In the 2019 movie Yesterday, the soon-to-be world-famous protagonist asks his manager, "Do I have to have an image?" To what she replies, "Well, if you don't have an image, then the lack of image becomes an image." I would add, then other people are going to craft your image without your input. That is what happened right afterwards. I won't go into detail, but IANIX and this blog are stories that are still referenced to this day as soon as the question of DNSSEC's usefulness comes up.

But people did not just get impacted by story alone. They got impacted by the story they would be inclined to hear. In a CircleID article I co-published with Roy Arends several weeks ago, Roy calls that DNSSEC fatigue, the kind of mood you get into after having been burned by early deployment, with the effect that whatever comes your way afterwards, you think you have heard it all before. This is negative bias due to personal experience, but actually, this is just in the ambit of a broader phenomenon. Indeed, humans do not grant their attention evenly and, for evolutionary reasons, are more attracted to drastic, menacing, and fearsome news items because these may be relevant for their survival. That is why people, even without any personal backstory involved, are more likely to react to negative storytelling and to let it stick. With the

effect that negative storytelling tends to pervade discourse lastingly, even after a situation might already be changed for the better. A bit like the colors of Holi, the famous Hindu festival celebrating spring's victory over winter. Peter and I played Holi last week, and as you can see, we have not fully recovered color-wise. That is because different types of color pigments get used. The bright and dry ones are stunningly beautiful and wash away easily. The dark and wet ones look a tad dirty and stick for days on your skin and your hair. They wear off on every garment you wear and stain it lastingly. In the case of DNSSEC, as late as 2022, DNSSEC-related articles were still likely to feature rumbling headlines, as the following sample shows. "DNSSEC claims another victim." "Another DNSSEC screw-up." "DNSSEC is notorious for breaking things." Just words, you might say, but actually consequential. In the aforementioned article, Roy Arends points out that the metaphor of the DNS camel, which progressively built up in the mid-2000s, effectively stopped innovation in the DNS realm for years. That is what stories can bring about, and metaphors are only star features of potent stories; metaphors can in themselves be or become condensed stories and act all the more powerfully because they are handy and can easily be passed along like coins.

But that is not all. I promised you emotions. There you go. I have already talked about DNSSEC fatigue, but there is a lot more. Let's have a look at that DNSSEC history again. It all starts out with the satisfaction of the developers. Nice. But then the good word gets spread, pride gets involved, and on both sides—pride of the

promoters, but also pride of those who are running their servers to the best of their ability. That is where things go south very quickly. Really, it is easy to embrace what people running servers will have felt when somehow forced to implement DNSSEC, they experience outages, not only distress, but downright face loss. As these outages become more widespread and get extensively listed, anger flares up on the server-running end. As a response to that, DNSSEC supporters experience shame, even though they assume flawed implementation or maintenance will have caused the outages. Some people jump on the bandwagon and angrily badmouth DNSSEC, but not just like that. As I was told by a senior security expert—we are going tabloid-style—the author of the most notorious anti-DNSSEC blog post was a former supporter who by some move became an outcast and thereafter turned against DNSSEC as if in revenge. This is now. From what I have seen since I started my work almost a year ago, there are a lot of former DNSSEC advocates, stellar people who seem kind of heartbroken about DNSSEC's unlucky fate, who have distanced themselves from it like you would from cheesy songs you were fond of in your teens, and who seem to feel a kind of shameful detachment that rather differs from adult equanimity, preventing them from speaking out in favor of DNSSEC even now as DNSSEC is about to be objectively mature.

What do we learn from this? Five minutes, okay. Human behavior is often irrational, even within communities that think highly of rationality and define themselves by it. Even there, there will be not

only facts but also hearsay and gossip, and all of that may bend people's assessment of factual matters and respectively hinder them in perceiving new developments which could alter their assessment. In short, story matters. People have feelings regarding other people but also regarding the fate of factual matters they care for because they work on it now or have worked on it in the past. Sometimes that will make them behave irrationally, or they might keep their feelings muted because it seems inadequate to have them in the first place. This, in turn, distorts how people will stand up for ill-fated devices even if they secretly care. In short, emotions matter.

Even in a fact-driven community like this one, human irrationality and people's emotions are factors that need to be taken into account or at least remembered when things go south and at best before they do. As a consequence, even technical matters need storytelling, so engage in it. And if you think that is distasteful, remind yourself that storytelling is not bad, it is just human. Your story does not have to be mad raving. It can very well be double-edged. Tell whatever story needs to be told, just do not leave it to others. Remember that even if you started it, the story will be told continuously with or without your input. Be alert to the versions that will be unfolding and react. If you face a problem like the one DNSSEC is facing now, odds are just working harder on the factual matter will not help, nor will more explanations. Whether you want to or not, you will have to engage in storytelling, and deeply so.

What does that mean for DNSSEC specifically? I will skip the details and just say that on the technical side, the course has been set for DNSSEC to be both accessible and smoothly operable. But for adoption to follow suit, as we have seen, the image issue at hand has to be tackled by providing updated information on DNSSEC's actual state-of-the-art, putting outdated publications into perspective, communicating in a balanced, non-evangelizing way, allowing people to make informed decisions about whether they should use DNSSEC, raising awareness on how narratives can impact the assessment of technical matters, and reaching out to the community by injecting lightheartedness and self-deprecating fun in the talk on DNSSEC, like in the PechaKucha I did. In short, it is about navigating this discourse on DNSSEC into more equanimous waters to allow for fresh, unbiased, and informed decision-making on a protocol about to reach maturity or, speaking with Holi, to leave winter behind.

Enough of DNSSEC. Let's have a quick look at DELEG now. Why conclude on DELEG in the first place? All the talk about DELEG has made me curious about how that will play out. From what I have gathered, DELEG is big enough an innovation for there to be something at stake, making it worthwhile to take a step aside early on and take a sharp look at the landscape before being on track at full throttle. Let me share some scattered thoughts about where DELEG could benefit from DNSSEC's hard-learned lessons to avoid history repeating. DELEG is optional in that you can always do things the old way if you want, like DNSSEC. This is not unusual, but

in the case of DELEG, this parallel operation mode is even explicitly stated as an expected state in the foreseeable future. This immediately raises the issue of deployment motivation. Remember the vicious circle? It cannot be any good because if it was, it would be deployed. Next, while the IETF draft displays a very level-headed tone, I have more than once heard people say things like, "But when DELEG comes," and you see where I am going. This is like waiting for DELEG. The risk here is double. For one, slip into a missionary tone that will unpleasantly remind people of how DNSSEC used to be hailed, and second, create a slackening of innovation in the wake of the big thing to come. Finally, there is timing. While designing a protocol obviously is a staggered process, do not actively encourage deployment before you have reduced complexity to a minimum by design, by documentation, and by outreach. You will have to... Sartre says, "Hell is other people," and Dante says that there are multiple circles in hell. I would encourage you to think about outreach to the community as the circles of hell. The further you go outbound, the more diverse the perspective becomes, and you will have to ask those people questions about what is their interest and what are the problems they really feel they have in order to get DELEG best shipped to the customer. Okay. I think I will conclude at that. Thank you.

PETER THOMASSEN Thank you very much, Barbara, for this interesting insight into storytelling and how that matters for technology standards. We do not have any time for questions left, but Barbara is here. A few

hearts are flying up on Zoom, I can see that, and we are available to talk and send us emails. Thank you for attending. The next session is in twenty-eight minutes.

[END OF TRANSCRIPTION]