ICANN74 | Policy Forum – DNSSEC and Security Workshop (1 of 2)
Monday, June 13, 2022 – 09:00 to 10:00 AMS

JACQUES LATOUR: All right, good morning. Welcome to the ICANN74 policy forum in The Hague or La Haye en Français. Today, we have the DNSSEC and Security Workshop. We have a half-day agenda, and then in the afternoon is Tech Day. So the DNSSEC and Security Workshop is done in cooperation with the Security and Stability Advisory Committee (SSAC). So we planned this workshop over the course, in between each ICANN meeting we have a committee and we plan the content of this workshop.

We have two sessions today. The first one we're going to talk about DNSSEC related things with Bruce from auDA, Johan from .se, and Yoshiro for NSEC3 stuff. And then in the afternoon we have a DNSSEC panel. Dan is going to do an overview of the DNSSEC deployment around the world. And after that we have a panel on DNSSEC DS automation. So I'm not exactly sure. We're going to figure out who's going to moderate that session when it happens. And that's basically it.

So the first presenter right now is Bruce. So .au zone split-key DNSSEC configuration. And he's here in person.

Note: The following is the output resulting from transcribing an audio file into a word/text document. Although the transcription is largely accurate, in some cases may be incomplete or inaccurate due to inaudible passages and grammatical corrections. It is posted as an aid to the original audio file, but should not be treated as an authoritative record.

BRUCE TONKIN: Okay, just to provide a bit of context with the .au country code domain name, previously the .au structure consisted of registration at the third level so that you'd have a domain name like example.com.au or example.org.au. So that the hierarchy within .au fairly similar to the, I guess, original DNS hierarchy with the .com, .net, .org, .edu, .gov sort of structure. So this year we changed to also supporting registration at the second level of .au.

Up until March this year we've outsourced the registry operator, the management of the central registry database to a supplier which is Donuts. And .au previously ran the top-level zone, the .au zone itself. Not unlike, if you like, ICANN running the root name space and adding names and making configuration changes through the IANA function and then the registry operators running things like .com, .net, etc. We had a fairly similar structure.

So by shifting to starting to do registration at the top level of .au, we also were shifting to having our registry operator publishing the top-level .au zone. Now previously at the second level, com.au, as is a fairly typical configuration the registry operator ran the DNSSEC system end-to-end. They have their own zone signing key and their own key signing key.

When we shifted to the mode where we'd have the registry operator also running at the second level of .au, we wanted to retain the control of the key signing key and maintain our relationship with IANA with respect to managing that key and have the registry operator operate with their own zone signing key for .au.

And so essentially this is a split-key configuration, and we found that this is not a configuration that is typical. Although it certainly is supported by the standards, most registries operate where it's the same organization that's running the key signing key and zone signing key.

So I've got a fairly detailed set of slides that will be published. I will just touch on a few points along the way but certainly let the audience dive into the detail if they're interested. That will be on the slides.

One of the first challenges we had was that the software that we were operating at the time, BIND, didn't really support this configuration where the key signing key is managed completely offline and it's just used to sign the zone signing keys. And then the zone signing keys are obviously used operationally. So we switched from using BIND to not Knot DNS for managing those top-level keys.

This also required us to upgrade the hardware signing infrastructure as well. So we were changing our policies and changing the software and changing the hardware. So almost an engineer's worst nightmare in that we're pretty much changing everything at the same time.

So we did quite a bit of work in just establishing secure methods for exchanging keys. Different organizations, different physically locations, and being able to exchange the zone signing keys and sign them with the key signing keys, etc. So a lot of testing involve, and a lot of the testing was focused on the procedures for signing and sharing keys and going through the full lifecycle over a year. So we typically update the zone signing keys quarterly and we update the key signing keys once a year. So we, unlike the root where there was a period where the key signing key wasn't updated, we update the key signing key every year.

And we built sort of a simulation environment to simulate that model. And again, mostly we're focusing on I guess what I would refer to as the key management processes. We're also running BIND on the production system, and in parallel we ran a test system that was running KNOT and ran with those two systems for quite a while.

One of the things we did before we did the transition from an environment where auDA was running the top-level .au zone and to transition to Afilias to operate that as we wanted to pre-publish the keys. So we got the new key signing key published and baked into the resolvers well before we actually did the change. And then when we registered the change, we reduced the TTLs down from 12 hours to 15 minutes to allow the switch between the keys.

And we did the actual cut. So when we went live with registrations for .au was on 24 March and we went live with the split-key DNS system on 9 March.

I know there was quite a bit of publicity. We did have an outage, and this was something that hadn't come up in our testing. It's one of those things when you switch to a production environment, we ended up with an issue. And essentially it was a software, and the software stopped publishing the RRSet records. And effectively that meant that if you were doing a full chain of trust from the top, the chain of trust would then break because we didn't have the signature records that were published.

And this was because we were incrementally publishing the zone. And we have a target of being able to make, if someone publishes or makes a change to the registry database, it gets published in the DNS within five minutes. And we have an SLA of something like 96% I think it is of all changes need to be visible in the DNS within five minutes which meant that the zone was continuously publishing. So an incremental update rather than a full zone transfer. And that particular software configuration resulted in the keys being missing and the DNSSEC signatures not being checked properly.

It took a while for that problem to materialize, and it was partly because we have long time to live on the signature records and most of the resolvers that were checking the DNSSEC signatures had cached the relevant keys and cached the relevant DNS records and they continued to operate. Also, a lot of the ISPs in Australia don't actually check the DNS signatures, so they weren't affected. So it was a very small percentage of people that noticed that the key was missing. We hadn't picked it up in our automated monitoring, so we have now put in monitoring systems to make sure that we can pick that up in future.

So once the outage was identified and the bug was found, what we then switched to instead of doing incremental updates every five minutes we shifted to manually signing the zone which we did once a day. So for about three days we manually published the zone once a day. So the outage that occurred was like a two-hour window when the signature records were missing. Once we identified what the issues were, we republished the zone. And then I think it was from Thursday until the Sunday of that week we were doing a zone update once a day.

So I mentioned the monitoring. Also mentioned the caching saved us in that the resolvers that were doing the checking weren't affected. We did find one public DNS resolver which was Cloudflare that the cache must have expired and they were trying to refresh the cache and users of the Cloudflare DNS resolver were the main users affected by the outage.

One of the things that was good about it was a classic case of international cooperation between teams. So we're based in

Australia. The Afilias Donuts team had a team of DNS people in North America. And they in turn were working with a team that had developed the Knot DNS software with the .cz registry.

And so the actual patch to fix the software problem was done within 24 hours. And then really we spent the next few days just testing that software thoroughly before we actually released it into production which was on the Sunday, three days later.

One of the big lessons with these things is making sure really whenever you have any kind of downtime in a system is then putting in monitoring to make sure it doesn't happen again. And so now we monitor to make sure that all the DNSSEC records are properly published in the zone every time the zone gets published. And we're further working on some software to make sure that the zone could…that it would be impossible to actually publish the zone if it was missing records. So there's further work going on there.

So hopefully I kept that to time. And back to you, Jacques.

JACQUES LATOUR: I think we have time for a question. Remember, you've got to put the question inside the Zoom Q&A pod. There's no…I don't think you're allowed to stand up to ask a question. Yeah, so everybody in the room, you should connect to the Zoom session.

KATHY SCHNITT: Jacques, we have a question from Brett Carr.

BRETT CARR: Hello. Bruce, have details of the bug been publicly disclosed so that anybody else who is using this software can be aware of it?

BRUCE TONKIN: That was a bit distorted because I couldn't quite hear the question. Sorry.

JACQUES LATOUR: Can you type your question in?

BRETT CARR: [inaudible] publicly disclosed [inaudible]?

BRUCE TONKIN: I'm sure that the features of the bug have been publicly disclosed. I think if you are using that piece of software, it's probably best to contact the software provider which is CZNIC. They published a patch that actually fixed that particular bug. It really would only occur if you're running a split-key configuration, and I'm not aware of anyone else who's doing that. But if you are intending to use the configuration that we're using, then it's just making sure you've got the latest version of the software. So they've now got a full production release where the bug has been fixed. So there's a patch available for the software we were using, and then there's a production version of the software available now too.

JACQUES LATOUR: So, Brett, could you type that question in the Zoom chat just for reference? Okay, no more questions, so next up is Johan. He's going to talk about .se outage due to DNSSEC. To you.

JOHAN STENSTAM: Thank you. Good morning. Do we have my slides somewhere?

UNIDENTIFIED MALE: They're just being set up now.

JOHAN STENSTAM: Yeah.

UNIDENTIFIED MALE: They are ready to go.

JOHAN STENSTAM: I still don't see new slides.

UNIDENTIFIED MALE: No? They disappeared?

JOHAN STENSTAM: And what I saw wasn't mine. Oh, these I recognize. Excellent. Many thanks. So I'm going to talk for a couple of minutes about an incident that .se had earlier this spring which is sort of I wouldn't say similar to the Australian experience but it's at least in the same ballpark as in DNSSEC issues and publishing zones that are not entirely correct.

Friday afternoon everyone was preparing for the weekend when we realized that we had a problem. The problem was that we had already published a version of the .se zone with a number of DNSSEC signatures that didn't validate, as in broken signatures were already published. This was obviously not a good thing, and we needed to sort that out as quickly as possible.

So what do you do when you have published a zone with broken signatures? You try to resign stuff. And resigning stuff in our case because we're using hardware HSMs in a cluster configuration means doing the same thing again, and that didn't work. And we tried to figure out different ways of getting this to work with no easy paths to a correct zone being found.

We thought about moving backwards to a previous version of the zone that was known good. But although that's practical or rather that's theoretically possible, in practice it's not really very useful when you have external providers and you have worldwide Anycast and lots of distributed versions of the bad zone. Because rolling back to a previous serial is time-consuming and complicated. So we really had to move forward. Next slide, please.

We had one HSM cluster and that HSM cluster has multiple so-called partitions. That's a mechanism inside HSM to separate one set of keys from another set of keys so that there is no way of them interfering with each other. So because we're running two TLDs, both .se and .nu, we had one partition for .se and one partition for .nu.

The strange thing here was that .se signatures were bad. Trying to generate new .se signatures generated new bad signatures. Then .nu stuff was working perfectly. So the HSMs in themselves were sort of working. It was just the .se stuff that resulted in broken signatures.

So we broke them apart and we tried to talk to one HSM at a time. That didn't work. And we tried to restart various parts. And in the end, the conclusion is that there was really no single HSM hardware failure here. The HSMs were continuing to do the right thing for .nu. It was just something very strange happening in the .se case. Next slide, please.

So the .se zone was signed for the first time a very long time ago. I believe it was the first ccTLD to be signed. And at that time, obviously the tool chains and the software weren't nearly as mature as it is today. Not saying that there cannot be bugs today, but it was clearly a more rough environment at that time when stuff was more immature.

That led .se to create a mostly homegrown zone generation and signing pipeline. Most of the parts have been upgraded over the years, clearly. However, there is—how should I put it—there's a difference between upgrading stuff, especially a zone generation pipeline for ccTLDs which has lots of components and lots of parts, and doing a completely new design from scratch. We had not done the latter.

So there were parts in the zone generation and zone signing pipeline that were very old. And when stuff gets older and it's really, really crucial for your business, there is a risk of falling into the trap of let's not touch it. It works. It's been working for ten years. It's perfect. Don't touch it. Next slide, please.

So that's where we were. And the obvious question now would be, don't you validate signatures before zone publication? And the answer is no. Or rather we didn't at that time. We do it now. As far as I understand, I wasn't there at the time, but as far as I understand the original reason for not doing validation of all signatures was simply time constraints. That would take too long given the zone publication pipeline.

And then again looking back at the don't touch it, it works fallacy, that issue had not been properly revisited when hardware was faster and the time constraints of validating all the signatures no longer being an issue which is the case right now.

Another part, not trying to justify not validating the signatures, is that the pipeline was designed to deal primarily with failing HSMs. That's why we had a cluster of HSMs and they were in high availability design and all that good stuff. So the scenario that the pipeline was designed to guard against was a failing HSM signaling some sort of error and the other one taking over, etc.

What happened in practice was that both of them failed at the same time without signaling any error. So obviously, all the checks and balances connected to the HSMs didn't save us in any way because we needed to have an outside completely external validation of the result from the HSMs and that part we were missing. Next slide, please.

As I said, we tried a bunch of different things. Breaking apart HSM cluster. Talking individually to the two HSMs, that didn't help. We restarted OpenDNSSEC which is what we use for signing the zone. We restarted the servers where we run OpenDNSSEC. Nothing worked and eventually we gave up on all other alternatives than actually physically rebooting the HSMs.

We had already broken them apart, so we took the first HSM and we physically rebooted that and then it started to work. That's good. And then for robustness and redundancy, also this being sort of scary and a late Friday night, the second HSM was restarted and that also came back. And suddenly we could generation a correct zone again. Next slide, please.

So the complete incident was fairly long. It was almost 13 hours. The reason why it was so long was that we actually did not initially notice that we had published a zone broken signatures. That took several hours. And once we noticed, we stopped publishing the zones. We ran with what we had because every test we did, every time we tried to generate new signatures or generate correct signatures where the previous signature was bad it just resulted in more broken signatures. So we didn't publish any new version of the zone from very shortly after we detected the problem until it was resolved. But still, it was 13 hours and at most as in the last broken zone that was published had more than 9,000 broken signatures published on the public Internet.

A saving grace to some extent is that none of the broken signatures covered any, let's call it, a delegation or a zone of national importance. This is sort of a sensitive matter because as a ccTLD registry, obviously every customer is just as important. And to the involved parties this was, of course, very, very bad. But still, the impact on Swedish society and the Internet in general was, obviously, smaller because of no really, really well-known zone being impacted. Next slide, please.

So it's been a couple of months and we've spent oceans of time on analysis, reports, future plans, what to do next, etc. And the short version of that is that the obvious thing of validation of the generated signatures, we added that the same weekend. So that was easy.

We've decided to basically toss out the entire zone generation and zone signing pipeline that we had which obviously has parts that are very old. And we are mostly done with a completely new design, and we will implement that after the vacations of the summer and take that into production use in the autumn. We will not keep any stuff from the old pipeline. We will do this from scratch. Next slide, please.

So what happened really here? What was the root cause? Well, the first thing to check given that we were using OpenDNSSEC as the signer with hardware HSMs for key storage, the first thing to check was OpenDNSSEC. There have been multiple independent OpenDNSSEC code analyses done both by us and by outside parties and also by [inaudible] labs, and the conclusion is that there is just no way that OpenDNSSEC can have caused this. It's just not possible.

The second thing to check was, of course, the HSMs. And there we are in a rather bad position because those HSMs were rebooted and both of them have worked perfectly since then. And because of that we cannot today reproduce the behavior. And as an HSM vendor, obviously what you want to do is you want to be able to reproduce the problem that the customer had, and they have not been able to do that.

So I fully sympathize with the problems that the HSM vendor has here. But still, we cannot find any other solution or any other explanation than this is a case of the HSMs misbehaving. There's no other way of explaining this.

And what makes it really, really concerning from our point of view is that I can basically live with one HSM behaving badly. It's, of course, nothing that you want to experience. You buy them because you want to not have problems. But still, single component failure, that can always happen, also with HSMs, and that's why you have multiple. And we did have multiple, and they both failed. At the same time. That's strange.

And it's also strange that while they didn't completely fail because the .nu stuff in the second partition was working perfectly all the time. So how is it possible that two separate HSMs can fail at the same time for the .se partition, which they had one copy of each, and not fail for the .nu partition? So obvious the HSMs are sort of okay, but the partition .se partition was sort of not okay for some reason. And how was that communicated between the HSMs?

And we haven't sorted this out. The hardware vendor is obviously trying to explain this still, but I do understand they have a problem. But it is a concern to us still. Next slide, please.

So the conclusion on our part is that the don't touch it, it works mentality was not good. That's dangerous. We have been running a signed ccTLD longer than most of the Internet, and that means that we have had more time to upgrade and replace than most of you. But my impression talking to other ccTLDs, etc., that have been asking about this is that it's not really unique to .se to have old stuff in their zone generation pipeline. I think that's something that is sort of if not general fallacy at least something that more registries do. And perhaps we should be slightly more aggressive about revisiting old stuff more frequently so that it doesn't come back to bite us at the wrong time. So that's something that we learned. We have to avoid letting stuff get too old.

The other thing is that we do have the opportunity because we don't have to publish all the time. We do not have any SLAs that say publish every five minutes or anything like that. We publish basically according to our own publish schedule which is once an hour. But if we have to delay publication, we will just do that. And it has basically no legal implications or anything like that. So being more prepared to hold publication in the light of any error is obviously good. And more monitoring to make sure that we detect any kind of error in the zone before publication is the way we're going. Next slide, please.

And I'm done. Any questions?

UNIDENTIFIED MALE: Thank you. This is a very interesting presentation. I have a question, but I can't put it in the chat because I'm a panelist.

JOHAN STENSTAM: But I can hear you.

UNIDENTIFIED MALE: What was the uptime of your HSM?

JOHAN STENSTAM: The two HSMs had different uptimes. They were not previously rebooted at the same time. And the difference in rebooting point in time was more than a month. They did have uptimes measured in multiple months, both of them. But they were not rebooted at the same time.

UNIDENTIFIED MALE: Okay, because many of us are running the same HSM.

JOHAN STENSTAM: I know.

UNIDENTIFIED MALE: Thank you.

JACQUES LATOUR: Peter has a question.

KATHY SCHNITT: Peter's question is, "Understanding the issue of lack of reproducibility but taking into account the critical infrastructure aspects including reporting and systematic oversight, to the extent you can disclose this, what is the role of the competent regulator in hunting the HSM bug?

JOHAN STENSTAM: Say again? What is the role of the…?

KATHY SCHNITT: The role of the competent regulator in hunting the HSM bug.

JOHAN STENSTAM: The only hunting has been done by us. The regulator has not been involved. We keep them informed, obviously, but they have not seen the need to, in your words, "hunt" the HSM vendor independently.

JACQUES LATOUR: That's it for questions?

UNIDENTIFIED MALE: No, we have one from Bruce. Bruce, go ahead.

BRUCE TONKIN: Just a comment about hardware [inaudible] with HSMs because it's sort of mentioned in the slides of our use but we also had a hardware failure in HSMs and similar reasons. They were old HSMs that had been bought at the same time and the batteries in them failed at the same time. So it's one of these things you don't normally expect. You think you've got redundancy and think one would fail and therefore you have time to replace the other. But we had two HSMs that failed at the same time.

And then our issue with the COVID supply chain. We're in Australia. None of the suppliers had hardware in country, and it was difficult to deliver. So part of the lesson, part of updating your hardware, I agree with that. Update it more often. But also try and get some time between when you update your equipment. If you update all the equipment at the same time, you have the same issue that if something goes wrong with some of that equipment, it's likely to happen around the same time.

JOHAN STENSTAM: Yeah, it's a bit like when we were young and we played with computer hardware and bought disks. If you buy disks to put them into [an arrayed] set, don't use all the disks on the same batch of disks. It is basically the same thing. So I sympathize with your problem with the battery replacement, but as far as I've been able to figure out we have replaced batteries on schedule in the HSMs and we have not replaced them at the same time. So they were not rebooted at the same time. The battery replacements were not at the same time. So while both of the HSMs are obviously several years old, we've done what we could to avoid having similar points in time that would cause something like this.

But still, the takeaway is any hardware, even in a cluster configuration, can break. That's why you need to have some sort of validation of the result before you publish.

JACQUES LATOUR: All right, thank you. Very interesting. Hopefully, we won't get too many. So we'll do the Q&A after Yoshiro, and then we'll do it there. Okay?

JOHAN STENSTAM: Yeah, thank you.

JACQUES LATOUR: Thank you, Johan. Next is Yoshiro talking about—from JPRS—are we ready NSEC3 guidance?

YOSHIRO YONEYA: Can you hear me?

JACQUES LATOUR: Yes, we can.

YOSHIRO YONEYA: Okay. So, [inaudible], can you please run my slides? Okay, so I'd like to talk about the new BCP for NSEC3 operation. I'd like to ask your suggestions how we can operate or introduce this new operational guidance. Next slide, please.

The background of NSEC3 guidance, it is Internet draft that is going to be BCP best current practice RFC very soon. It is in the RFC editor queue now, so I hope it will be published in a few months.

For more technical background of this proposal, please refer to Viktor Dukhovni's talk at the ICANN70 DNSSEC workshop. Also, please refer to the draft itself. I don't dive into the details of this proposal, but I'd like to talk about the impact of this draft or this BCP RFC. Next slide, please.

So the NSEC3 guidance affects both zone publishers or authoritative DNS side and the DNSSEC validator or full resolver side. But the time of when they will follow this guidance may differ. Due to this timing difference it possibly may cause name resolution failure of TLDs. That is very large outages of the Internet. That is very highly concerning. So I'll talk about it, provide some mitigation proposals. Next slide, please.

So the objective of this talk is to explain the possibility of the large outages at TLDs and propose some mitigations. This is not against NSEC3 guidance, but I'm aiming to smooth deployment of this guidance. Next slide, please.

The key points of NSEC3 guidance are that it is [inaudible] that NSEC3 guidance indicates that using iteration count larger than 0 is less effective and possibly security threat because it can be a cause of DOS attack to both the authoritative and the resolver side. Next slide, please.

The NSEC3 guidance proposes for the zone publisher side that if NSEC3 must be used at the zone operator, then the iteration count of 0 must be used to alleviate computational burdens. The recommended NSEC3 parameters are: SHA-1, no extra iterations, and empty salt in the regular format: IN NSEC3PARAM 1 0 0 -. The use of opt-out based NSEC3 records is not recommended except for very large and sparsely signed zones. Next slide, please.

Key point for the validators or resolving sites, there are two [great] recommendations. Validating resolvers may return an insecure response when the process NSEC3 records with iterations larger than 0. Or resolvers may also return a SERVFAIL response NSEC3 records with iterations larger than 0. Next slide, please.

So this has a very large impact to the TLD operation using NSEC3. The DNS name resolution of TLDs who are using NSEC3 with iteration count larger than 0 may be resulted in insecure or SERVFAIL some day after the publication of this NSEC3 guidance BCP RFC.

As far as I observed during the IETF discussion, major DNS software or service developers are favorable to this guidance and therefore default setting for DNSSEC validator will follow the BCP in the future. I'm not sure when, but in the future. Next slide, please.

For that reason especially when a large public DNS resolver such as Google or Cloudflare started to follow this BCP, TLDs who are not following the BCP will be possible to become unresolvable globally. This is fear of a large outage of TLDs.

If this happened, customer support or ISPs will be overflowed by claims from the end users. And customer support is hard to know the root cause of this failure. And after several minutes or hours the validator operators will put the TLDs into NTA, negative trust anchor, permanently. And this is a negative practice for DNSSEC deployment, I think. And when they put the TLDs into NTA permanently, it is very hard to recover from this situation. Next slide, please.

This slide shows how many TLDs will be affected. [I list] TLDs whose NSEC3 which uses iteration count larger than 0. There are about 1,150 TLDs in total [as of earlier this June.] And this data source is TLD Apex History. You can find the data from the ICANN website. Next slide, please.

This is a proposal for avoiding large outages of TLDs. At the TLD side change the NSEC3 parameters to recommended value of NSEC3 guidance as soon as possible prior or soon after the publication of this BCP RFC. At least iteration count to 0 and empty salt is highly recommended. And the completion of changes is desirable within a half year after the BCP RFC publication. Next slide, please.

At the validator or full resolver side prepare a certain grace period before changing the treatment of name resolution for iteration count larger than 0 to insecure or SERVFAIL. At lease prepare a half year grace period after the BCP RFC publication. And if they're willing to change to SERVFAIL, staged approach that change to insecure first for a certain period and then change to SERVFAIL if preferable. Next slide, please.

Third is the community side. Let have a global consensus regarding to a certain grace period prior to validator side's changes. So I think about how to have a global target date. So I'm not sure if the next DNS Flag Day target and date are decided already, but this would be a good candidate, wouldn't it?

So this is all of my presentation. Next slide, please. So I would like to hear your suggestions for how we can prepare for the NSEC3 guidance. Thank you.

JACQUES LATOUR: Thank you, Yoshiro. This is interesting. Do we have any questions?

KATHY SCHNITT: Wes, you can ask your question.

WES HARDAKER: Thank you. Thank you for this work and thank you for looking at the [problem] space with respect to TLDs. I will note one conflict of interest statement. I'm the author of the RFC that's about to be published, so take that with a grain of salt. But I want to make sure that people when they read the document—we talked about this a lot in the DNSOP working group—that it shouldn't be an immediate transition for validating resolvers. Specifically, there's a sentence that says validating resolver operators and validating resolvers software implementers are encouraged to continue evaluating NSEC3 iteration count deployments but lower their default acceptable limits over time. In other words, not immediately. We need to continue doing these measurements over time to figure out when the right time is for validating software, both deployment and operations as well as defaults for software to change over time. It's not supposed to be immediately RFC is published that all validating software should go to 0 immediately.

YOSHIRO YONEYA: Thank you.

JACQUES LATOUR: Any other questions?

KATHY SCHNITT: Jacques, we have a question from Warren in the chat. It says, "But I don't think that any of the TLDs violate the NSEC3 guidance, do they?"

JACQUES LATOUR: Well, I think the list that Yoshiro provided, they all…can you go back to that slide?

YOSHIRO YONEYA: About this?

JACQUES LATOUR: So the list. Yeah, this. Potentially, all of those are not meeting the guidance, right, from the BCP?

YOSHIRO YONEYA: Yes. After the publication of the BCP. They are using iteration count larger than 0.

KATHY SCHNITT: Jacques, can you remember to turn off your mic when you're not talking, please? Warren, go ahead.

WARREN KUMARI: So there's actually a follow-on to the question. That's only if you assume that the validators are going to start treating 0 as the number that they're not willing to accept. And that's somewhat of a pathological case, right? Like all of the resolvers are accepting much larger numbers. There's a list in the BCP of a number of large public resolvers, and they're all treating 100 as the number. So if resolver implementations decided to be silly, you could potentially have that affected. So I think it's not how many TLDs would be affected. It's how many TLDs could potentially be affected if people did really dumb things.

JACQUES LATOUR: There's one question in the chat from [inaudible]. Do you want to read it?

KATHY SCHNITT: Sure. Question regarding the .se tool chain. Why do you believe that new software is more stable than old one? New software does not contain several types of bugs simply because it does not provide the associated features. But old software has the experience with a lot of edge cases the new software did not see yet. The only reason to replace old software is that the developers are dying.

JOHAN STENSTAM: Interesting point. I think the very short answer is that…let's call it inspection of the old software has convinced me that replacement is the right thing. I'm not claiming that new software is perfect. But what we're talking about here is not so much third-party software like a new version of Knot DNS or a new version of some tool. But we're talking about the actual infrastructure that maintains the integrity of the zone generation and the zone signing pipeline. You can do that in…I think we have more experience today how to do this in a better way than we did 15 years ago.

JACQUES LATOUR: Thank you. Any other questions for any sessions? We have two minutes to go. No questions? Thank you. So we have a 30-minute break?

KATHY SCHNITT: We have a 30-minute break, yes. We can meet back here at 10:30. Thank you.

[END OF TRANSCRIPTION]