ICANN70 | Virtual Community Forum – Tech Day (2 of 4)
Monday, March 22, 2021 - 10:30 to 12:00 EST

KIMBERLY CARLSON: Welcome back to the Tech Day Part 2. Again, my name is Kim Carlson. Kathy Schnitt and I will be your remote participation managers. Please note that this session is being recorded and follows the ICANN expected standards of behavior. We will put any other reminders and housekeeping items in the chat. So we will go ahead and turn this back over to you. Thanks, [inaudible].

EBERHARD LISSE: Thank you very much. Ulrich Wisser from the Swedish Internet Foundation, you have the floor.

ULRICH WISSER: Yes, hello. Kimberly, are you showing my slides or am I?

EBERHARD LISSE: It's coming up just now.

ULRICH WISSER: Ah, yes. Okay, thank you. Hey, hello, everybody. Today I wanted to talk about the transition from NSEC3 to NSEC for the .NU zone. This is obviously not a downgrade. NSEC3 and NSEC have the exact same functionality. They are just two versions of the same functionality. Next slide, please.

Yes, so my name is Ulrich Wisser, and I work for the Swedish Internet Foundation. Besides .SE we also run the .NU ccTLD. I have worked for the Internet Foundation 13 years now. I have been a software developer for the registry system, and I'm the co-chair of the CENTR-Tech Working Group and a member of the DNS-OARC program committee. Next slide, please.

So why would we want to do this? Well, we actually took over the management of the .NU zone in 2013. Taking over a zone is a lot of change to happen, and so we didn't want to add also DNSSEC change on top of that. And so we did continue to run DNSSEC with the exact same parameters that it has been before, and that's why we did run the zone with NSEC3 and at the time we did run with RSA/SHA. Next slide, please.

So when we…three years ago now we changed the zone to ECDSA but we continued with NSEC3. And that is basically because we didn't know how to [inaudible]. Then we started to look into this, how to do the transition from NSEC3 to NSEC. If you read the RFC—that's obviously always the first thing you should do—the RFC is very clear. You remove all the NSEC3 records, you put in NSEC records, and you're good to go. Next slide, please.

So what could possibly go wrong? Yes, we don't know, is the answer. And that is a very bad answer—next slide, please—if you have these people looking over you. The first one, PTS, is the telecom regulator in Sweden. They actually have a say about how we run zones. And you have the Network and Information Security (NIS) Directive of the EU. We

have the CEO of the foundation who has actually some requirements of how we do our job. And then we have, of course, our CISO who thinks that we should even do DNSSEC in a very responsible manner and I tend to agree with her actually. So we had to have a little bit better answer than what could possibly go wrong. Next slide, please.

So we did some testing. Next slide, please.

Then if you want to test this, there is actually a little problem with this because NSEC is actually the answer that says there is no answer. So how do you actually test that you got the correct answer that there is no answer? This somehow gets a little bit philosophical, but I think we managed to do some checks on this. Next slide, please.

So we set up a testbed actually where we would set up the same zone on two name servers, one running with NSEC and one running with NSEC3. And then we had a proxy in front of it, and the proxy would then proxy to one of the two instances depending on actually the label names that came into the proxy. So you had a no data answer with wildcards, without wildcards, name errors, all these. The standard kind of things when NSEC records are used we checked, and this went very well so no question.

Then we thought, okay, let's see about the corner cases. So you ask for something that doesn't exist, but we don't send you any NSEC records. No NSEC, no NSEC3, nothing. So what will resolvers do? Or we send you both but only one of them will cover the label or none covered the label. And of course now we're already in a very strange land where which name server would send NSEC and NSEC3 records at the same time. It

was really strange setup, but we wanted to see how far can we push this. Next slide, please.

So we got the list of public resolvers from the list on this webpage here. Next slide, please.

So these are the results that came out. As you see, we did test 71 servers, 32 are obviously not validating DNSSEC and 39 are validating DNSSEC and we didn't have any NSEC records. They would just say SERVFAIL. Pretty standard behavior. I can say in all the other cases where we send NSEC or NSEC3 answers they would have totally standard [inaudible]. There was absolutely nothing to report.

And then in these strange cases afterwards you see that if we come to this nothing covered the label, then it's not really what we would expect. But on the other hand, it's an NXDOMAIN answer and we [were happy]. So overall we were really surprised by this but also it gave us the confidence that this is going to work. Next slide, please.

So that really looks good, so we're good to go. Next slide, please.

So we make a test run. Next slide, please.

Yes, so what we did is we took the .NU zone, made a copy, changed the name to .NUTEST.NU, signed it with OpenDNSSEC because that is what we use in the daily operations of the .NU zone, put it on one Anycast provider, and then we made a measurement with RIPE Atlas probes. Next slide, please.

So here you see the results. We asked for random domain names. In the upper slide you can see what outcomes we got back. We got mostly NXDOMAIN, some strangely NOERROR, and we got absolutely no SERVFAIL back.

And then in the lower graph you can see we actually counted the number of answers we got with NSEC or with NSEC3 records. And, okay, there's a large amount of resolvers that do not do DNSSEC and so we don't get any NSEC records. But the ones that do, you see that in the beginning we only get NSEC3 and when we change to NSEC then it gradually goes down and we go to NSEC records. Exactly the behavior you would expect from resolvers. Next slide, please.

So that looks really good. Exactly the results that we expected from the whole thing, and so we declared this good to go. And even all the other interested parties that I listed in the beginning saw that, okay, this looks good. You're good to go. Next slide, please.

So now we did the same thing with the .NU zone. We used OpenDNSSEC but we have three different Anycast providers. And now we measure not only with 500 RIPE Atlas probes but with 5,000. Thank you to RIPE NCC because they made this possible. There's some magic [inaudible] that you have to do to be allowed to use 5,000 probes. Next slide, please.

Yes, so this is the result we got on [all the days] where we did the transition. This looks a little bit different than the other one, and we were really surprised by this. You see we have a large number of NXDOMAIN answers. Nothing strange here. We have these NOERROR again, and then we have absolutely no SERVFAIL which is quite good.

And then in the lower graph here again we counted the number of answers with NSEC and NSEC3, and here suddenly there are a lot of answers with NSEC or NSEC3, any of this. But then suddenly when we do the transition from NSEC3 to NSEC, the NSEC answers go up a lot and the NO NSEC answers go down a lot. It's like, wow, a lot of resolvers suddenly started validating because we went to NSEC. That doesn't seem really likely. So we actually did an investigation into this.

But on the other hand, what you see, we didn't get any SERVFAIL, and so we declared this transition a success. We had absolutely no complaints from any parties in the world about the transition. Nobody couldn't find any domains they were supposed to find or anything. Next slide, please.

So this was a success. When we looked into this little glitch there, it actually turned out this was a miss with the RIPE probes. The RIPE probes, if you don't set anything yourself, the RIPE probes will actually set an EDNS buffer of 512 bytes. So the NSEC3 answers basically didn't fit into the result and that's why we didn't get any. That was actually a little bit of a miss, but overall we are really happy with the transition. Everything went according to plan and was a success. Thank you. Any questions?

EBERHARD LISSE: Thank you very much. The obvious and [trivial] question is why did you do this? It's obviously because you would run .SE on NSEC and you don't want to run two different systems, isn't it?

ULRICH WISSER: Yes, we always have run .SE on NSEC. And actually since 2017 we publish the .SE zone and the .NU zone. So they are publicly available, and so there is really no need for any hiding/disabling zone walking in the zone because you can download it. So you already have access to the zone. We didn't need NSEC3. In the last two years, there have been several registries in Europe who had problems with the NSEC3 chain. So we don't have a need for NSEC3 and we see that other people have operational problems, and so NSEC3 really looked more like a liability than an asset. And that's why we wanted to change to NSEC.

EBERHARD LISSE: And as I said, obviously one would prefer to run one system [for the two].

ULRICH WISSER: Yes.

EBERHARD LISSE: There's a question from Yoshiro Yoneya. Roughly how many names were under .NU at that stage? Since you publish the zone, this is a totally [inaudible] question.

ULRICH WISSER: Yeah, it would be around 400,000 domain names, between 400,000 and 500,000. It depends a little bit on which time of the year you do this, but that's approximately the amount of domains in the .NU zone.

EBERHARD LISSE: Thank you very much. Are there any other questions? We are very good for time. Okay, now there is the question which I think does not belong to you. Why is there not [inaudible] on public-dns.info? Since you don't…unless you operate that website, I don't think that's a question that you can answer, or can you?

ULRICH WISSER: I think the website lists open resolvers, and actually it lists the big public resolvers that actually you should be able to access, that want you to access them, that offer service to the public. And then they have a large list of, I would say, involuntary public resolvers. Maybe some home routers that are open to the public and stuff like that. I really didn't feel comfortable running tests on people's infrastructure that haven't given their permission to do so, so I didn't look at all that part. I just took the list of public resolvers that are publicly available and are intended to be so.

EBERHARD LISSE: But to answer the question, there is a link on that website. So, Ulrich, if you really want to know, click on that link, contact the operator of the website. And if you get a nice answer, maybe send it to us so that we can publish it on one of the lists.

ULRICH WISSER: Yes. I later found another list that APNIC operates and had I known the list beforehand, I obviously would have gone with the APNIC list, but you know.

EBERHARD LISSE: Yeah. But it's not really a question for this, but it's a good question in itself.

ULRICH WISSER: Yeah.

EBERHARD LISSE: Mark Elkins asks, did the zone size increase, and do you know how many .NU domain names in the zone are signed? In other words, what percentage of names in the zone are signed?

ULRICH WISSER: Yes. We didn't run opt-out for NSEC3, so the zone didn't really increase in size for the NSEC records. There is a little bit over 40% of the .NU names [that] are signed. That is actually—are we good on time still?

EBERHARD LISSE: Yes, yes, yes. I will [inaudible] [if we move on]. Take your time.

ULRICH WISSER: Yes? Okay. So .SE was actually the first TLD in the world to be signed with DNSSEC, and so we are kind of really into DNSSEC. And we have a program for our registrars where they get a 5% price reduction on domains that are DNSSEC signed. That's why we have actually a lot of domains signed. The .SE zone is over 50% and the .NU zone is around 40%. And we actually are actively working on driving this up.

EBERHARD LISSE: Yes, that's one way. That's the easiest way of doing it is giving the registrars an incentive.

ULRICH WISSER: Yes.

EBERHARD LISSE: Okay, we have two and a half minutes left. So Rubens Kuhl asks, was the publishing process interrupted during change from NSEC3 to NSEC?

ULRICH WISSER: Nominally, yes. We stopped publishing the zone, and then we obviously ran the process internally. So we usually publish the zone once every hour, and we actually managed to produce a new NSEC-signed zone and run all the tests on it before the hour was up. And so we could just then wait for the next publishing procedure to publish an NSEC-signed zone and we didn't miss a single zone update.

EBERHARD LISSE: That's a good point to know if somebody wants to do that. One can then for the day this is going to happen change the publishing interval accordingly so one can test how long it takes to sign it and then change it a little bit so that [you do it], you run it, you publish the new one. And when that is then stabilized, a day later you do it back.

ULRICH WISSER: Yep.

EBERHARD LISSE: All right, thank you very much.

ULRICH WISSER: Yes.

EBERHARD LISSE: Quite interesting [inaudible], I must say, even if going against the grain. But it makes all sorts of sense, and I liked the little images produced and which illustrate the thing very nicely. Thank you very much again.

ULRICH WISSER: Thank you.

EBERHARD LISSE: Okay, now we have got Mark Robertshaw from Oxford Information Labs. Thank you very much for putting it up so that I didn't have to look it up myself. [inaudible] contacted me first, and that's why we got into

each other. I had never heard of them before, so I didn't know what OXIL stands for. Mark, you have the floor.

MARK ROBERTSHAW: Thank you very much. Good morning, good afternoon, wherever you are in the world. I'm going to speak today about RDAP. I'm Mark Robertshaw and I'm the CTO of Oxford Information Labs which is a U.K. based tech company. We specialize in cybersecurity and security solutions generally. And the last eight years or so we've been providing the technical services and the platform for a U.K. focused registrar which is called Netistrar. We're going to talk today about the implementation of RDAP for the registrar called Netistrar and how we got on with that. So if we could go to the next slide, please, Kimberly. Thank you.

So I guess to start with, what is RDAP? Many of you will know what RDAP is already, but it's the Registration Data Access Protocol. That's it's full name. This is from the ICANN website. It's an eventual replacement for WHOIS for accessing current registration data for domains in particular.

Importantly, it's a standardized and validatable data format using modern web standards. Unlike the legacy WHOIS standard which is a lot more text based, this is now using a much stronger data format using the becoming industry standard JSON format for that. I think equally importantly, it's extensible. So the data structure allows for future enhancements, including secure access going forward. Okay, so the next slide, please.

So obviously one of the things about RDAP is it's something that's actually being, I guess in a sense, imposed upon us as registrars. We have to implement it. There has not been a choice about that. However, taking a positive slant on it, I'd like to see it as a real benefit to us as registrars as well. I note some very strong benefits toward moving to RDAP, as mentioned, from the WHOIS standard.

I think the most important aspects are it's well defined and a rigid data format. As mentioned already, unlike the very text based [inaudible] format of WHOIS which was susceptible to lots of error and lots of misinterpretation or misimplementation by different parties, RDAP is a guaranteed and rigid format for data interchange.

I suppose a corollary to that is it's easily validated using third-party tools which are available online. And so testing your RDAP solutions becomes a lot more straightforward than perhaps guaranteeing that your WHOIS data would be reliable across the board.

And I guess very important for us as technical people, it's implemented as standard SSL 443 web service which makes deployment of RDAP and its use cases in terms of cacheability and use in CDN environments a lot more attractive than the legacy Port 43 WHOIS service.

And I guess also for the future, RDAP affords a lot more opportunity to provide much better access to data over time, including being able to implement various different levels of security and it has built in provision for that going forward. So next slide, please.

To give an idea of the RDAP data structure, as mentioned before, we've come from a legacy environment where WHOIS is very much a flat [inaudible] format with key value pairs in a very unstructured manner. We've now moved to a much more structured approach which models much more closely the kind of [programming] paradigms that those who like to develop data structures would appreciate.

So obviously we have the domain in the middle, our key entity. And modeling very much the workflow of the domain name system, we have obviously one or more name servers attached to a domain. And [inaudible] we have the entities here which are essentially contacts which would model our registrant, our admin, billing, and technical contacts. And they themselves use a very standardized vCard format for encoding the data for that. So we have a very object oriented model here within the RDAP structure.

Other things to note are this use of these other meta items—events, notices, and remarks—which can be attached to any of the entities above. They can be part of the domain, the name server, or indeed the entities. And events provide a lot more structure and a lot more flexibility around the reporting and querying of key events on a domain name.

For example, a typical event would be a registration and other such. Also, last changed/last modified date it also captures as events within the RDAP structure. So you end up with a lot more structure and a lot more flexibility in terms of encoding that.

Likewise, notices and remarks are ways in which we can encode in a very standard some of the stuff that we're required to publish as part of the structure such as our policies and our meta data but also provide the information about whether our data structures are complete or whether they've been subjected to some sort of data redaction for whatever reason. So there's built in structure within these events, notices, and remarks structures to actually be a lot more explicit about the data that we're producing and all-in-all gives us a much better overall shape or structure than we've had from the WHOIS system. Next slide, please.

So moving on to our implementation of RDAP a little bit, talking about how we got on with it when implementing for Netistrar, I think as mentioned key to us was the event tracking for assets. It produced a lot more rational output for us and provided a very helpful way for us to track our data better.

Another very important point is there's built in support in the RDAP format for GDPR redaction and for privacy/proxy which is something that has been a bit of a thorn in the side of registrars over the years in terms of how best to handle [custom] privacy/proxy provided by registrars but also to since the GDPR redaction a couple of years ago how best to deal with redacted data and what is the correct standard for that.

RDAP actually has the facility to mark GDPR data up and to be clear about whether your data is complete, actually has some gaps, or whether you're using a custom privacy/proxy.

And as mentioned in the previous slides, there's rational encoding of the various things that we're required to produce as part of our ICANN accreditation such as our policies. And using the JSON format which is these days a very established format for web services, you can provide very standardized links which can be followed and queried to link to our polices and is in a sense a self-documenting system in that sense.

If there's one negative to implementing RDAP as a registrar, I would say it would be the adoption of the vCard format. This is a very verbose format for encoding data, and actually it comes from an established standard. But I'm aware that across the industry I've heard rumblings that many people have found the introduction of this in the RDAP spec is somewhat verbose and clumsy. But that's the only negative point I would want to make about our experience with implementing RDAP. Other than that, as mentioned above, we see a lot of the benefits of using the RDAP format in terms of encoding data in a much more rational way. So next slide, please.

Again, I guess the final point is deployment of our RDAP solution. As mentioned, RDAP [implements as a] standard web service. This means it slots into existing web server clusters without any headache at all. And in actual fact, for ourselves we've found that implementing WHOIS has been a bit of anomaly for our system. Most of our system is built using modern web standards for our dashboards, our APIs, everything else that we provide as a registrar. And we've been left with this dangling WHOIS service which has had to be handled in a very special way.

Actually, RDAP is a breath of fresh air because actually implementation is exactly the same as it would be for any of our other JSON based APIs and so, therefore, has been very straightforward to slot into our cluster and, therefore, benefits from the usual load balancing and other techniques that we use generally for our systems.

I guess equally important is that because it's a standard web service, we can employee third-party edge network protection too much thought. Many of the big names out there—we use Cloudflare, but many people use different similar sort of services—they're very much geared toward using traffic on Port 443 or Port 80 and do not have any sort of protection out of the box for custom ports. Therefore, the fact that RDAP has been deployed as a standardized web service on Port 443 means that we can immediately use the DDoS rate limiting that is offered by our edge cache but also to be able to aggressively cache without having to do both strategies for ourselves.

So from a deployment perspective, RDAP has been a breath of fresh air. It has been a new service that has just slotted into our existing infrastructure without too much disruption of services and has really been an [additive] service in that sense to us. So the next slide, please.

I was hoping to be able to give a quick demo, so let's see if the next slide comes up for us, if that's okay.

Okay, so be able to show you how simple it is to actually use an RDAP service, you can see that this is our own domain we're querying via our RDAP service. You can see it on a regular browser. You can see here the data coming back. Very rational JSON format. You can see that we're

starting to see the contact data coming through here. In the middle you can see the use of the vCard there. We've actually got a status removed, which is actually as mentioned earlier on the use case for a GDPR redacted record. As we go down, you can see that we're got the remarks and you see in the remarks there a result set truncated due to authorization and tells you that actually the data is being truncated. So there's lot of self-descriptive encoding within the RDAP structure. You can see again further down, the general notice at the bottom as well that we actually have built in provision for encoding our policies and the [inaudible] complaints form and the various things that you're required to do in a very natural and very self-describing way. So hopefully, that has given you a flavor for how RDAP works and also a sense of how it worked for us implementing on our systems.

And I think the final slide now. I think I'll be very happy to take any questions on that if anybody has any.

EBERHARD LISSE: Thank you very much. I had to find the unmute button. I am personally a great fan of JSON if only that I had opportunity. Most of you know my day job. I'm a gynecologist, and I've been able to analyze 30,000 pap smear results with easy, widely available tools. And it convinced me that JSON is very [inaudible] format.

Personally, I don't mind vCards. I would, for example, like to have one of our mailing lists that we have at the ccNSO a contact list in case of if we have issues. I would like to have that in that format so I can just put

it in my cellphone and I can just push a button. But this is from just a user perspective.

Rubens Kuhl had a question. Let me read it. Some edge network protection services get confused with JSON API responses such as RDAP instead of HTML code. Have you suffered any such issues while deploying RDAP?

MARK ROBERTSHAW: Not so much. I mean, generally, we've not experienced that problem very much. Provided you set the content type headers correctly on your responses, most of the edge caches we work with will obey what you send back and shouldn't disrupt flow. So it hasn't been a problem for us.

EBERHARD LISSE: Any other questions? Just one. Is this in any way publishable? Like open source or libraries that others can use or something?

MARK ROBERTSHAW: At the moment, our platform is being developed in a fairly proprietary manner. But in a sense, the RDAP format is very self-describing, and I'm sure we'll see very soon some open source libraries appearing around this whole area because they're very straightforward to develop.

EBERHARD LISSE: There are some around. I just wondered. This is a question that you may have noticed I ask everybody that comes [inaudible]. All right, I cannot see anymore questions. Going, going, gone. Thank you very much, again, for an interesting presentation.

MARK ROBERTSHAW: Thank you very much.

EBERHARD LISSE: Next will be my compatriot Benno Overeinder from NLnet Labs. You have the floor.

BENNO OVEREINDER: Thank you. I think I've put on the camera. Welcome, everyone. This is a presentation of work done by my dear colleagues Berry van Halderen and Roland van Rijswijk. So they should get all the credit here. Am I on camera? Because I don't see myself, but I hope so.

EBERHARD LISSE: The screen is shared of your presentation, so you're not visible. You're screen sharing your presentation is.

BENNO OVEREINDER: Oh, okay. Excellent. That's fine, thanks. Thank you. So this is a presentation about a project we executed at NLnet Labs. It's about DNSSEC key ceremonies. Let's go further and then explain what we have done. Next slide, please.

So this is kind of a good audience and maybe also maybe not the intended audience, although I would love to hear your feedback. Most of the audience here do know about DNSSEC and probably also run a TLD, either a ccTLD or a gTLD. DNSSEC has seen a wide uptake in the ccTLD and the gTLD.

At your right, you see the picture of all the ccTLDs signed and the status of them. As far as I have checked, and it's also policy, all the new gTLDs and actually all the existing gTLDs, the older ones, are also DNSSEC signed. So that's good.

And also, if you sign your zone with your key, you also are aware that the keys are really valuable because if your key is compromised, other versions can impersonate you and can impersonate the operator or the zone owner with all [inaudible] and damage as a consequence. So you want to protect your key material. Often this is done by using an HSM. So you keep your protected key material in an HSM and keep it safe. So that's a practice done by many TLDs, so that's not exceptional. But then we go to the next step. Next slide, please.

HSM is not connected to the Internet quite often also because it contains high-value material. You want to protect it further, also to put HSM in a so-called air-gapped environment. Air-gapped, think of a bunker, a locked room. There's also a physical barrier to get to the HSM.

But it also brings some complexity. Namely, how do you interact with the HSM with your DNS system? We call this interaction, how do you introduce a key, how do you bring a key out of the ceremony, the ceremony. Think of the IANA ceremony. We have put here a picture of

one of the ceremonies, a snapshot of the ceremony with public witnesses. Also, other TLDs have implemented such a ceremony, and that's important.

The thing is that you have to decide if the ceremony is purely a technical process. So it is for security safeguarding of your material, so it does need some well-defined procedures. But also, it's trust important. You want to have public scrutiny and transparency to build trust in your DNSSEC system.

And you also have to plan in this situation for all possible scenarios. Think of ICANN's blog last year. It was called [Conducting a Key Signing Ceremony in the Face of COVID-19]. So we have to improvise if travel is not possible. And I have learned that [GPRS] had a similar fallback plan implemented last year to sign their zone. So if you think about ceremonies, you have to think about these different scenarios. Next slide, please.

Although DNSSEC and implementation and deployment of this DNSSEC is widespread deployed, there's no common approach in doing so. Especially with tooling there's no common toolset. So it would really help to define a standardized guideline for DNSSEC signing. That can help the community to implement the secure ceremony with appropriate tooling.

With the ceremony requirements and design you have to think of when do you generate the ZSK, for example. Do you generate ZSKs online, bring them into the secure environment—into the air-gapped environment, the bunker—sign them with the KSK, and export them?

Or do you generate the ZSKs in the bunker and only export them? All these kinds of requirements. Another one is what are, for example, the features of the HSM? We have to think about them before implementing the ceremony.

So standardizing these requirements should also result in a more secure ceremony and help automate this process. Here on your right you see this ceremony description and the documentation, kind of a blueprint of the ceremony we've published on GitHub as part of our project. Later I will give a URL. You can find all the information we have published. Next slide, please.

As just mentioned, better standardization allows for easier and better automation. We call the automation here a recipe. A recipe is a number of steps, sequential, simple, verifiable that goes through a ceremony from one step to another step. And that can be executed in an air-gapped HSM or an air-gapped environment.

So a recipe we have developed exists of a number of preconditions—which keys are present in the HSM, for example—a set of steps, fixed order. There's no need for control logic, so there's no IF-THEN-ELSE or a loop. This is intentional because then you know there's no infinite loop. You have a verifiable number of steps you can check. And the result of the execution of the recipe should be a cooked outcome, a dinner. And we discuss all three steps in the next slide. The details of how you make a recipe are also documented on the GitHub page snapshot taken here on the slide. The URLs, etc., can be found later.

And finally, we also as a proof of concept implemented this we this toolset and integrated with OpenDNSSEC signer. We'll come back to that later. But think of this. We have some IDs taken from OpenDNSSEC, but it's really designed to be portable and interoperable with other solutions also. Think of BIND or [inaudible] [name server], for example, Knot DNS. Next slide, please.

Cooking the recipe. It's important to note here, and I already mentioned, the recipe is a fairly straightforward set of instructions. There's no complexity in the recipe because it should be straightforward to execute in a secure environment. So that's important. I'll come to many more details later in the next slide. Sorry, one slide back just [sketching] what a tool does, actually.

So we generate a recipe. We have tooling for generating recipes. We have, of course, processing the recipe. We call that cooking in a secure environment. And with the result, we want to export it out of the secure environment and incorporate it in the operational environment. Next slide.

So this is a recipe. This is the input for our cooking. The tool is OKS. And if you call OKS cook, then it takes a recipe. On your left side slide, you see a recipe. It defines a number of things like the keyset you want to sign, the key you want to sign with, until it's [valid], for example. You see all the parameters on your left side.

After cooking this recipe, you get the output. You get the cooked recipe. I shortened the left side. So the head of the right side is actually the same—actionType: produceSignedKeyset—because that's the most

important function of the tool actually, to generate a signed keyset. But you see that on the right side there's one section added—cooked—and that's actually the keyset. The signed keyset you can import, for example, in a name server.

So this is how you cook a recipe. How do I generate recipes? So that's the next slide, please.

Producing the recipe. Recipe producing can be quite complex. I'll come to that later. But it's important that the recipe can be produced entirely beforehand. So you can do a dry run, for example, the day before. If you want to invite observers, you want to have a perfect run of your recipe. There's no hiccups and everything is in the correct order. The day before, for example, you can do a dry run outside the secure environment and check if all the steps are properly executed and the result is as intended actually.

The tool also because we say all the complexity is in actually creating the recipe and not in executing because in creating the recipe, that's outside the secure environment. You can validate the recipe and then you have to validate this recipe, execute it in the secure environment. But creating a recipe is complex because you have to generate all these steps without any control logic.

Here you see an example, for example, to have a pre-generated, preproduced recipe. At the right top you see how the tool OKS can produce a recipe. There's a configuration file below that, quite familiar, that defines the policies in the signing of the key signing policies. There's, for example, the validity of the signed keyset which is one

month, a refresh every 30 days, etc. So this is quite familiar also with other tools.

They have also a so-called KASP. BIND, Knot DNS, and also OpenDNSSEC, they have this ID of a KASP. So this is quite a generic ID you can use. With this KASP you define the policies of your key and you give that as an input to the OKS produce command, and then it will generate keysets for the next year, 2021, for this year.

The current prototype we have implemented implements actually a full set of features, only we have now also because of time limitation focused on a number of—well, we call it—a number of ceremonies that are most common. Namely, that you have the KSK and the ZSK generated in the bunker and export the ZSK to the outside of the bunker. So we don't support yet, but that can be done later. We had to make some choices, but you could also in the future generate your ZSKs externally, import them in the secured environment, get them signed in the HSM in the secured environment, and export them back. So functionality is [full], but what we have implemented in the [script] is this structured ceremony. Next slide, please.

Probably a little bit ahead of time, but that's good for discussion. After producing a recipe and cooking a recipe, of course, we have to eat our meal, consume the resource. So a successful ceremony will result in a cooked recipe that contains multiple sets of signed keys. This different set of signed keysets need to become active over time.

How is the export and import actually in the operation environment done? First, we use the OKS consume command. You can use it, and

then all new keys that have been generated are imported into the HSM just at once. And the second time, for example, you see the command below that, the second one. It's consume with a specific date, and that will produce a signed keyset for that specific time. That is very useful. It makes it very suitable to run, for example, daily in a [chron] or in a daily job.

And the output can be integrated. We implement it with ODS, OpenDNSSEC. So the output of this OKS consume with a specific date is a file, and you can easily integrate that with a BIND or a Knot DNS or with OpenDNSSEC. We tested with OpenDNSSEC. And OpenDNSSEC has a signer enforcer component, and so the output actually of this toolset, OKS consume for a specific time, actually replaces the enforcer. So with integration of ODS you only have the signer and actually this toolset that replaces the enforcer and tells the signer what to do with signing the zone.

Actually, I think I've told everything. Yeah, again, the concepts are generic. We designed everything to be interoperable and to be [integrable] with other tools, popular name resolvers like BIND and Knot. Next slide.

So future work. Get your feedback, of course. I think many people here attending the session have had experience with signing keys, also having implementing key ceremonies. I'm happy to hear their feedback, their experience, how they implemented their key ceremony, what we can learn from each other.

This is for the future, of course. Also, if there's sufficient interest in the toolset and standardizing the key ceremony, [does it then] make sense to have this recipe API taken to an IETF or something other like standardized on this so we can have some industrial common interface to implement key ceremony toolsets and having a standardized blueprint? Of course, everybody implements it differently, but with the blueprints you make a number of decisions that are still within the template where you can design tooling for. And of course, also getting other open source DNSSEC developers interested in this toolset and implement integration with their tools.

That's it, actually. Further reading, I think—oh, next slide, please.

Yeah, so all the background information and links to toolsets and documentation and blueprint, etc., can be found in this blog. There's a blog post describing the toolset on a high level, but in the URLs you find in the blog post you find all the details if you want to use the toolset. Both using the toolset but also how recipes are written, how you can generate them, etc.

Yeah, looking forward of course—well, final slide—but you can leave this slide up. Of course, very interested in your feedback. I'm open for questions and comments. Thank you.

EBERHARD LISSE: Thank you very much. I have two screens. The cursor didn't want to move to the one where I could unmute myself. These little things.

Jaromir Talir, one of the panelists, had a question. You have the floor, Jaromir.

BENNO OVEREINDER: Please, go ahead.

JAROMIR TALIR: Thank you, Eberhard. And thank you, Benno, for the presentation. You had the same presentation at the CENTR Tech workshop and I promised, or I discussed this, how this could be potentially implemented in [Knot] DNS. After a quick analysis, it looks to me that it's maybe just a terminology thing that what you call the recipe is the same what is called key signing request in other offline key ceremonies. And the cooked recipe, the opposite is the signed key response. Is this assumption correct?

BENNO OVEREINDER: Yeah, indeed, especially for the output, for sure. Yeah, that's similar to what you said. The first one, can you repeat? The recipe was…?

JAROMIR TALIR: Well, that's the key signing request, the list of the DNS key resource record that will be online over the next, like, half a year. This is the terminology from ICANN that they use for the root.

BENNO OVEREINDER: Yeah, okay, thank you. Yeah, definitely. So, yeah, you are correct. [So that] makes sense to link this terminology with what we have use, yep.

JAROMIR TALIR: Okay, thank you.

BENNO OVEREINDER: Yeah, thank you.

EBERHARD LISSE: There are two questions. Let's start with the first one. Is this tool applicable for both ZSK/KSK or ZSK only?

BENNO OVEREINDER: The tool is applicable for both. Also for KSK. So all for a key rollover, definitely. So I think one of the examples it was both a KSK and a ZSK had to be signed with an already referenced KSK. In the tool what will happen is that in the bunker with existing KSK already [in the] HSM a new KSK will be generated, signed, and a new ZSK or set of ZSKs will be generated and signed and exported. So it's also for KSK key rollover.

EBERHARD LISSE: Thank you. Angela Matlapeng from .BW. Thank you, Benno, for the intriguing presentation. My question is, what would suit more and in what use case between SoftHSM and HSM. Also, out of curiosity, are you looking into providing DNSSEC as a service for automatic key rollover and management in the future.

BENNO OVEREINDER: Yeah, the last one is a very interesting question. So the first part of the question, the decision between SoftHSM and real HSM is also, of course, what are the requirements of the organization. Does it need FIPS compliance? Does it need to be FIPS [4], for example, or not?

But we are aware and we know of organizations that use SoftHSM but on a laptop not connected to the Internet. So it's not a real HSM, but it's SoftHSM so it has PKCS #11 interface so it acts as an HSM. It's running on a laptop disconnected from the Internet in a bunker, so it's physically protected, etc. But it's not FIPS [4] compliant, for example, but it has a number of additional security.

So again, I know that organizations have their own decisions, make their own decisions, the security officer to take the required boxes. And in some situations given the requirements, SoftHSM implementation on a laptop fulfills all the requirements but still needs to be offline, for example. And that's perfectly implementable. So for this toolset it actually can work both, with HSM or SoftHSM.

I hope I answered your question. It was a little bit longer answer than I intended.

EBERHARD LISSE: The second question are you looking into providing DNSSEC as service for automated key rollover and management in the future?

BENNO OVEREINDER: No, we as an organization are not thinking of that yet. But I know it is quite interesting, can be an interesting service indeed. But we are software developers. We provide tools. We do also run our own software. But running this as a service also requires different organization or some steps into operations which we haven't done today. But I think it would be in general an interesting take, yeah, interesting solution for DNS operators to run this as a service to send your zone as a service to a DNSSEC signing solution and get the results back. And of course, also having some tools, etc., to configure your policies, etc., yeah. I hope this….

EBERHARD LISSE: [inaudible]

BENNO OVEREINDER: Please go ahead.

EBERHARD LISSE: No, no. Carry on.

BENNO OVEREINDER: Okay. No, I was just curious of this answering the question or some other questions are triggered by my answer.

EBERHARD LISSE: I personally think I wouldn't look at something like this as a service. [inaudible] clearinghouse established and other nonprofits that are

trustworthy. I'm not [inaudible]. I'm not saying that this is…I like the idea of generating your own policy and if there was a secure way of transmitting it to the [inaudible] when the ceremony is run that you can see that you basically put the screen on a Zoom thing or something like this.

The idea about a laptop, get three or four cheap laptops, glue the ethernet port shut with superglue. I don't know how to physically disable the Wi-Fi. I don't think you get laptops without Wi-Fi anymore. But if you get one where you can superglue the ethernet and it doesn't have a Wi-Fi, you put it in tamperproof envelopes in a bunker, that's as good as it can be whether it is FIPS compliant or not.

BENNO OVEREINDER: Yeah, indeed. So Berry, one of the project leads, was thinking of as a summer project to have a number of Raspberry Pis and then disable everything except for the USB, for example. Glue everything in some epoxy or whatever so you cannot get to the hardware anymore and something like SoftHSM installed in a kind of, of course, not a FIPS compliant HSM but something like an HSM. No Internet, only a USB port, etc. And then also limiting the kernel that it's only listening to the USB and very restricted, something like that, so a kind of [stripped] Linux kernel. I think it's still cool to do and make it as a giveaway, not pretending it's a real HSM. But it is doable, exactly as you mentioned, Eberhard.

EBERHARD LISSE: You could even put tamper proofing in. I remember a presentation a few years ago. You can put the Raspberry little plate into a special box which is tamperproof and has maybe a battery or something that will fry the [RAM] or whatever or the [ROM] where these things are stored if somebody tampers with this. Raspberry and these kinds of things are well suited for this.

BENNO OVEREINDER: Yeah, I know. There are solutions, indeed, yeah. Maybe Rick Lamb? Oh, yeah, thank you [inaudible]. Rick Lamb also made a presentation once about a self-made HSM. It was very funny.

EBERHARD LISSE: That's the one that I was mentioning. He mentioned he did it in a cooking oven in the kitchen or baked something. But that was what I remembered. If you can put a tamper proofing in, if I don't know…I'm a gynecologist. As I say, I'm a baby mechanic. I don't really know about these things. But it could be easy and it's relatively cheap and you can do them [inaudible].

BENNO OVEREINDER: Indeed, yeah.

EBERHARD LISSE: And it's probably cheaper than doing a laptop. There is now another question here. What kind of requirements do we need to apply DNSSEC and RPKI on .NI? I don't know what that means.

BENNO OVEREINDER: Oh, to apply DNSSEC and RPKI. Yeah, I don't know. Maybe Nelson can a little bit elaborate.

EBERHARD LISSE: Let's take this offline, Nelson. Email to either on the list or to Benno or something and we'll follow this up. We can always publish this on lists.

BENNO OVEREINDER: Definitely, yeah. Because both are relevant, of course, in security and also protecting your assets, DNS and your routing assets. And RPKI is indeed signing your resources, your IP blocks. Okay, yeah, I will contact you, Nelson, and we can discuss that later offline.

Also, I see on the chat Mario sent as a question. Would it support a double KSK key implementation? For that question, I have to really defer to Berry to be fair. I think all the tools, all the possible scenarios are implemented. But as you hear, I formulated it this way, that definitely Berry knows. But we implemented most of the common scenarios, but in principle the tool can…we have to extend the tool then to also implement double KSK scenarios, yes. But for now, we have a limited set of scenarios because time limitation. We want to finish the project by the end of [next/previous] year, so we had to make some decisions and went for the most common case. But it's extendable and we can…and this kind of input is very useful so we can think it over industry or user requirements and if we can implement it in the second phase. Thank you.

EBERHARD LISSE: Thank you very much.

BENNO OVEREINDER: Yeah, it was great fun.

EBERHARD LISSE: Interesting presentation, indeed. I like these toolkits that are useful for smaller ccTLDs but especially that we have got more than one or two options and that are available free of charge so that people can choose what they want to do.

BENNO OVEREINDER: Yeah, definitely. Indeed, it is really intended for everyone. And we also—maybe I should mention this—we also did look at the ICANN and also at PCH. How they implemented their key ceremonies that you referred to, Eberhard. PCH is also providing DNSSEC services to their customers. And we also had contact with PCH and in principle our toolset can also implement these well-known ceremonies. Okay, thank you.

EBERHARD LISSE: Thank you very much.

BENNO OVEREINDER: Thank you very much. Bye.

EBERHARD LISSE: I'll give now the floor to Iliya Bazlyankov about ccTLD security practices.

ILIYA BAZLYANKOV: Thank you. Good morning, good afternoon, and good evening, everyone. My name is Iliya Bazlyankov. I am with EDOMS from Bulgaria. We are consulting small or medium scale ccTLDs that run their own platform or an open source one that is already available. During our work we noticed some common mistakes that people tend to make, and we created a small security framework or just a checklist that we distribute to everyone that is interested. And I want to share today a few of our practices that we have developed through the years. Next slide, please.

Why it's important for ccTLD to be secure. First, you need to be reliable. You are hosting the infrastructure of a whole country—banks, military domains, government domains. You don't want in case of a security accident such websites to disappear or worse. You don't want somebody to violate your database and modify some data and steal money from banks' customers or create phishing websites for government services like was happening recently.

You also want to avoid data leaks of sensitive data of customers, and you want to avoid DDoS that could break your systems temporarily. It can kill your WHOIS server or other systems. DDoS is not fatal for a ccTLD because DNS would continue working, but all other services could be down. Next slide, please.

Our checklist is divided in four categories. They are specific, backups, hardware, and personnel. Next slide, please.

Our software category, first we start with checking the server security of our customer from the very basic secure terminal access, VPN, database, [inaudible] for public access, limit web services, [open] ports, to more advanced fine tuning components for speed and for closing some features that can be exploited by attackers.

Regarding software components, we advise to find the balance between stability and updates. It means that you need to have a schedule for updates and regularly update your operating system, [inaudible] libraries to new version. I've seen a ccTLD running [one to ten and third layer] software which is long ago not anymore supported. And it has a lot of security vulnerabilities, and now it will be really hard to update to a newer server.

What else? For registry components, I will talk in a bit. Generally, I would recommend any public login panel to be secured with two-factor authentication, at least an SMS or [Google] authenticator or any other service it can work in your country. Next slide, please.

Basic division of registry components. the WHOIS, the RDAP server, and registrar billing should be available to public. Which means we recommend them to be on a separate machine at least with most the ports closed and open only the web service parts. They should connect to your server with a database which is closed [from abroad]. Our recommendation is your database to be fully closed [from abroad]. You don't have public ID. You will connect by internal IP, internal network,

or VPN. Just do not expose your database to the world. I've seen publicly open phpMyAdmin to a registry system which was not good.

After you set up a separate machine with your WHOIS and RDAP servers, our recommendation is to have a copy of your database for them. In this case, if there is a DDoS to your public services, your main database will continue to be functional. Then on a separate machine behind firewall you put all your registrar services like EPP, registrar panel, or domain availability system. You restrict them via IP only to your registrars and to the registry staff.

And as I mentioned, the hardest security should be for database, also for zone generator and DNSSEC. Just do not expose them publicly because these can be compromised not only by persons but even by bots that go and scan your [whole] IP networks. Next slide, please.

Regarding backups, I cannot stress how important it is to have backups. Again, I have seen systems without backups or with backups that are a month old which when you register domain names in your registry, you cannot afford that. You should have a backup also in a remote datacenter with [inaudible] different [inaudible] different country.

As an example, I can give earlier this month a datacenter in Strasbourg burned and some of the local backups were not restorable. Imagine your ccTLD database being [inaudible] datacenter and you don't have a backup. So everything happens, and you should plan and make a good backup strategy.

As a small compromise, I can also advise to look into the escrow format. If you are a small ccTLD, you don't need to deposit to an escrow provider but you can store at an external server the XML escrow files. In this case, in case of disaster you will be able to restore your registry objects quite easily. Also for TLDs with more than a few hundred domains, we recommend to build hot and cold standby systems in the same or in different datacenters that you could switch to them in case of disaster without affecting your registrars. Next slide, please.

Regarding to hardware, if you're running your own datacenter, you should know how access should be restricted. You should have a special category of employees that have access to the datacenter. If you choose a public datacenter, make sure that they have some sort of certification. Or if it's not available in your country, just talk to them what are their security practices. Next slide, please.

Regarding personnel, more and more phishing attacks could happen and somebody that wants to gain access to your registry can steal the email, the credentials of a remote member of your staff. Then it can steal server credentials if they are stored and encrypted and then gain access. So my advice is to train your employees how to handle sensitive data credentials, certificates, or anything that is used in your security policy.

Also, when you let some of your employees go, before that remove their access because it could lead to problems not only with registries, with other web services, even Facebook pages. Next slide, please.

Thank you for your attention. I'm always available at that email if you have any questions, if you need any help. I can share that checklist with you. In the meantime, I'm available for any questions if you have them. If not, thank you.

EBERHARD LISSE: Thank you very much. It's always good to have a security presentation. Even though we have had them before, it's always good to refresh this. Remember last time we had [inaudible] present their situation when they were seriously attacked and their business logic infrastructure was attacked which was for them a bit of a problem.

I personally don't think you need a hot and cold standby for a small ccTLD. You need to have a backup that is automatic. It needs to be tested, and you need to have either a machine next to it which you can put your backup in or you have to run a system in parallel. But for a small ccTLD it's good enough if your machine goes down that the [maintenance] continuous backup can be restored within a day.

And it is more important to practice this than to invest into complicated setups with hot and cold standby and then if the DNS is not reachable it automatically switches, that kind of thing. On a small registry, if the system falls down, the DNS is usually not affected, caching is happening. And if you can get it up within a few hours to a day, the worst that can happen is that one or two domain names that were in the process to register need to be manually sorted out. But as I said, it's much, much more important to have a plan than to have a convoluted plan.

And then the other thing which is very important which I fully agree with, you need individual logins for everything. Every staff member for everything, whether it is advertising on Facebook, whether it is manipulating the database, should have their own login so that login access can be blocked to individual and to IP addresses. And when somebody goes, that access, the passwords can be removed or changed to something totally else so that nobody who leaves the entity can take their credentials with them.

Therefore, it's also whether you do [ISO] or not it's important to document this so that you exactly know if somebody leaves, that's the process they need to do. Chop, chop, chop, chop. Exit interview, credentials, and so on.

I don't see any questions in the Q&A, and I only saw a comment in the chat which wasn't a question. We are a little bit early, so if somebody wants to ask, please raise the hand or ask something in Q&A. Okay, thank you very much.

ILIYA BAZLYANKOV: Thank you.

EBERHARD LISSE: I see this is not the case. Thank you very much. It's really good to have a good summary. It's also important to note we publish all of this on the ccNSO [website]. So these summaries are always very good. You can download them and go through your process and see, have I done

everything? Have I overlooked one thing? Tick the buttons and cross the Is and dot the Ts, as they say.

Again, we have another break now. Can you put the second slide, please? The second [inaudible]. I made a mistake. I used first and second. This is the second break. We will be back in exactly 37 minutes at 17:30 UTC.

**[END OF TRANSCRIPTION]**