JOHANNESBURG – Tech Day (Part 1)
Monday, June 26, 2017 – 13:30 to 15:00 JNB
ICANN59 | Johannesburg, South Africa

EBERHARD LISSE: Good afternoon, everybody. Welcome to Tech Day at ICANN 59. I don't know whether it's the 30th or the 28th or something. Doesn't really matter. Welcome, everybody. My name is Eberhard Lisse. I am, as you know, the ccTLD manager of .na and I chair the technical working group that organizes this venture.

On the policy meetings, we only have half a day, and we do this always in conjunction with the DNSSEC workshop. They do it in the morning, they sponsor the lunch which we then participate in, and then we do the afternoon thing.

In the morning, we have got basically two sessions. If we run late, we will skip the break. If we run early, we will have a longer break so that we get into the schedule again. The first presentation is usually the host presentation. David Peall from DNS Africa, .africa, will basically do their thing. They're not [ZABNA] but they have so many acronyms, I always get confused.

Then, Fred Baker is here from [ISE.] He's supposed to be here. So, if necessary, we'll just switch the things a little bit around. Then we'll hear from our colleagues in Botswana. Kamanga is

there, I don't know which one is the first surname. Moakofi is the first name. And then we'll have Amreesh talk about the DNSSEC infrastructure at AFRINIC.

Then, Matthew Zook will talk a little bit about the TLD analytics. Dave Piscitello will introduce the tool formally known as the Domain Abuse Reporting Tool. Then Linda Müller from Knipp will talk about their version of DART that they have developed. It's basically a commercial thing, but I have convinced them to make it available free of charge to smaller ccTLDs, and we have a tradition to allow presentations like those on Tech Day.

Then, Francisco Arias will speak about the ICANN monitoring system API. This is basically something developed for gTLDs, but I would be very interested in seeing what it can do and whether we can maybe use this as ccTLDs as well. Not necessarily on ICANN infrastructure, but the idea sounds good.

And then Alejandra Reynoso from .gt and I will talk about a few issues of possible compromise that we were advised of. I'm finding out that some changes we made or didn't make had unintended consequences. Fortunately, nothing happened, but we were advised about credible threats, and so we thought we'd bring it to the attention of you guys in case you have something similar brewing that you can check it out.

Theo Kramer, who I don't see at the moment, will then close up. If he doesn't pitch up, we'll find somebody to do it for him. That said, David Peall will do the first presentation. One thing I need to remember, we'll have questions after each presentation. Every question needs to be preceded by the name of whoever asks because the remote audience don't see us, and the remote audience, if there are questions, will get preference.

DAVID PEALL: Good afternoon. Thank you for giving me the opportunity to share some of our experiences. I'm also very interested in feedback from the community in terms of what we've done. I'm David Peall, I'm a Systems Architect with DNS Business for DNS Africa or DNS Pty Ltd, depending on the day.

The first section – so, I've got two presentations, one after each other. The first one is on our DNSSEC implementation where we've tried to create a resilient DNSSEC system. And the second one is on a live migration of a gTLD using [with] DNSSEC sign while it is signed.

So, who is DNS Business? We are a registry service provider. We do not operate TLDs in isolation. We always partner with and provide services to a registry operator and provide a platform, technical solutions, and technical guidance to the registry operator.

The system started developing in 2010, so it's relatively new. And because we had the value of hindsight, we were able to create the system knowing that DNS implementations for EPP varied across the spectrum. Even though EPP is a standard, the way it's been implemented by different registries and the policies behind those registries are often complex and different to each other.

So, from the onset, we started with the idea that the registry software would be separated from our policy. So, while we do deploy different policies, it's always the same piece of code running our registry software. [inaudible] likely then support ccTLDs, [gLTDs,] gTLDs and brands because of that policy dynamic nature.

The policy is a database entry for us, so we are able to even deploy policies live in a running system without any downtime. This has great advantages in terms of our [SLAs] that we provide our customers. So, that's just the summary of what DNS Business does for its customers.

I'm going to go into phase 1 of what we were trying to achieve with a resilient and stable DNSSEC infrastructure. I think a number of people have followed this route as well. It was our first step. And the target was really to create a solution that wouldn't leave us in the lurch if something broke. So, we created a redundant system with database replication for the software

from one site to the other. So, obviously, these are operating in two different, distinct datacenters.

The databases for the signing software is key management. It maintains the timing around the keys. It maintains the link between the keys and the HSMs. The second step is the filesystem replication for the HSM. The HSMs we use are the Thales, and they do not keep the keys inside the HSM, so they store the HSMs on an NFS share. It's a bit of a [cheek] if you consider what you pay for the device, and you have to then synchronize that filesystem yourself.

The third step is to make sure the HSMs can both read the same set of data. This involves pulling out your operator cards and administrator cards and the groups that you've set up, and synchronizing the primary or the master hidden key for the whole HSM system. In the Thales-specific implementation, they call this the security world.

So, in a nutshell, you have two datacenters with the same software, the same HSM implementation, and the same data for [assigning] zones. This however left us with some concerns, and I'll go through those. The first one is that replication – and anyone who runs a database will know that replication does not equal backup. If you corrupt your source, you corrupt all your

slaves. So, you need to obviously maintain regular database backups, and the same with filesystems.

With backups, there's always the chance that you would lose data. Databases have streaming backup, but there's always a time or quantity of data that is determined when each backup is happening, and with filesystems, I'm sure the situation could be even worse.

So, there's always the chance for – a slim chance, but there's always a chance for the loss of data. That made us uneasy. The next thing that we had was that the signer on the standby system could not be running at the same time as the signer on the live datacenter, and this is because they would fight to manage keys in their own database.

So, to test the system, you'd have to actually shut down your primary server, your primary signer, bring up your standby server to test it, which we thought was a terrible solution and fraught with potential dangers.

The third problem with this setup is that you are vendor locking. In other words, because you are maintaining a key infrastructure between two datacenters, they have to be the same hardware, potentially even the same model and make, to be able to achieve this setup. And thirdly, then you've also got your software vendor locking, because obviously the database is

specific to the software you're using, and for them to operate in this manner, they would have to be the same software version and vendor.

So, then we looked at this and we thought about, how could we do this better? We went with DNSSEC resilience in four easy steps. Two independent signers operating in their own datacenters, each using their own software, their own hardware, own versions. There's no requirement that those signing systems have any relationship with each other.

The DNS records for both systems are maintained in the parent or root, just as you would with a single server system. So, instead of just the single DS record and your parent, you'd obviously have two, one for each of these systems that you are signing. And the way that this is achieved, in terms of making it work, is that we take the DNS keys from each signing system and we enter them into the source of the unsigned data for each opposite system. I've got some diagrams that may help explain this. And then the result is that you have two systems signing a zone, both validate if published, without any requirement to wait for [TTLs] or propagation.

What'd you do to my slide? Okay, well, you get the gist of it.

There's registry data there on the left. This is the typical DNSSEC implementation. Registry data goes into the zone file. The zone

file's in XFR, AXFR, IXFR. I just use XFR for the generic term. Signer receives this data, does its magic, and out comes a signed zone. You can just read that signed zone at this point without any risk. Once you're happy that this is working, you can put your DS record for the signed zone in the parent. That'll obviously anchor the chain of trust to the parent, and your zone is now signed from the point of a validating resolver.

So, the next step would be to set up your second signing system. Totally different system. Could be anywhere you choose. Preferably obviously in a different datacenter, same or different software, same or different hardware, and the process is very much the same.

Once you have got your DS record from the second signing system, you can give it to the parent. However, a TLD, like a ccTLD, you would often have to wait for the next step. Otherwise, IANA will reject that second DS record until the next step.

Okay. So, now what we've done in this is we've actually taken the zone generator and queried the DNS keys from the opposite signing system. This is a simple thing to script. You basically do [inaudible] with the DNS key type, and you get back the list of keys that you need to put in your unsigned zone for the opposite system.

So, both signing systems are generating a zone. Each signing system generates a zone that has got potentially all four of the – at least four of the keys, a zone signing key and a key signing key from each system. In the first system, that data set of DNS keys is signed by its own key signing key, so its RRSIG will match its own key signing key. That RSSIG is obviously valid because its DNS record is in there in the root, so that your chain of trust is maintained.

Similarly, it's a zone signing key and its RRSIGs generated for the authoritative data would also validate, because its zone signing key is signed by the KSK that is validated in the root. Exactly the same applies for second independent signing system. It signs the DNS key set with its own key signing key. That key signing key is anchored to the root with its own DS record, and its own zone signing key signs its own authoritative data.

What you end up with is two zones with the same information, but completely different sets of RSSIGs, both of which are valid at all times. You can then simply failover between your two distribution servers to choose which DNS signer or system you need to use. The failover is a matter of a simple configuration of where the data comes from. Questions?

EBERHARD LISSE: Any questions?

ROBERT MARTIN-LEGÉNE: Robert Martin-Legéne from PCH. [Mark] tells me to be kind. You did mention something about a key ceremony that I think is probably part of this, because you mentioned something about the KSK already having signed the DNS keyset.

DAVID PEALL: Sorry, would you repeat? A ceremony?

ROBERT MARTIN-LEGÉNE: At what point does the KSK sign the DNS key [RZ]?

DAVID PEALLE: Each of the signing systems will sign the data that comes from its unsigned source which will contain the other system's DNS keys, as well as the signer will input its own DNS key data and sign the whole complete set with its KSK.

ROBERT MARTIN-LEGÉNE: Okay, so the KSK is active in this system at the same time as ZSK is active?

DAVID PEALLE: Both systems will have an active KSK and ZSK at all times.

ROBERT MARTIN-LEGÉNE: So, the KSK is live in this system? That's my point, it's basically online in the system.

DAVID PEALLE: Yes. We don't have an offline signing system. We sign while the HSM is there and open.

ROBERT MARTIN-LEGÉNE: Okay, and the KSK, you have one KSK? Or you have two then? You have two DS records in the root?

DAVID PEALLE: Two DS records in the root, two KSKs, two independent systems. One can completely fall off the planet, it doesn't affect the other system.

ROBERT MARTIN-LEGÉNE: Okay. Good. Thank you.

DAVID PEALLE: Okay, so I thought that would go too quickly, so I added something else. Can I get the next slide? It's not working.

Okay, so this is a bit of a case study. We migrated from one RSP to another, it's a [gTLD,] and obviously it's signed, so we had to maintain this signed zone during migration.

I believe they're just fixing the slides. Okay, so next slide, please. Can I get the second slide? Thank you.

So, when migrating between registry service providers, ICANN allows a certain amount of downtime for your EPP and WHOIS. It's also not good for your clients to have that offline for too much time, and that's something I'm not going to deal with in this talk.

The second thing that we're going to deal with is the DNS and DNSSEC [are not allows] downtime at all according to the ICANN SLA. And even if you're not under that SLA, it's a very bad thing to have your DNS break. Having your DNSSEC break breaks the DNS for all clients using validating resolvers, so it counts as broken.

Next slide, please. It's the arrow on the far left. Arrow down. Yes, thank you.

Alright, so I'm going to take you through a real-world example of a migration that we did with a TLD, and I'd like to first start off by saying that this migration was a cooperative migration and would not have gone as successfully or as easily if we didn't

have the cooperation of the then-current registry service provider. Obviously, you can transfer a zone without the cooperation of the current registry operator or registry service provider, but it doesn't make life any easier.

So, on the 21st of March, the server generated a KSK and ZSK for the zone .wien. We did this by just sending a dummy zone into our signing software, low SOA so it wouldn't interfere with future zones that we gave it. And that gives us a KSK and ZSK that we then provided to the then-current registry service provider to include in the unsigned zone.

Their signer then obviously signed this key data. You can see the theme here, it's really a repetition of what we achieved in our resilient DNS infrastructure. Once we confirmed that those keys are visible in the zone, we went to IANA and asked for the DS record for our KSK to be added into the zone.

Then we set up a slave server for the .wien zone, and we slaved the then-current RSP's copy of the zone to our proposed nameservers that we were going to use.

Next slide, please.

Okay, so this is a layout of the situation at that point. We've got three nameservers. The nameserver set on the left is not active, it's got the live zone. Our own distribution server which is

receiving XFR from the current registry service provider. Their DS record is in the root zone, and we have provided DNS keys to the current registry service provider to be signed.

Next slide, please.

So, then the next step was to include our name servers in the zone file, so we contacted the registry service provider, asked them to add NS 1, 2, 3. And then we confirmed we have a couple of confirmations, updates [earlier] had been applied from IANA. Then we request a new update once the nameservers are in the zone. This update will include our proposed nameservers. At this point, they'll be the current RSP's nameservers and our nameservers. It's a live set of nameservers, but all holding zone and all serving the zone.

Then we asked the current RSP to remove the nameservers from the zone, so the zone file itself only contains the new RSP's nameservers. Their servers still serve the zone from the point of view that the root will still direct all queries to the complete set. Then we request that IANA remove those current RSP nameservers, and then we confirmed that everything was completed on the 14th of April.

Next slide, please.

So, this is the situation at that point. We have our DNS keys and nameservers in the zone from the current RSP, who gives us the signed data back to our distribution server to our nameservers, which are answering all queries for the zone. We have anchored our DS key in the root. Obviously, the DNS key set is not signed with our key signing key yet, but this is the situation we had before going live.

Next slide, please.

On the day of migration, so we left it like that until we actually did the registry service provider migration, and this had a window of 12 hours for services not DNS. But once we were happy with the data that was coming out of the last [escrow import] we then changed the setup where the current RPS's DNS keys were included in our unsigned zone, that unsigned zone that we generated from our own dataset now was transferred to our signing system, which then signed the then-current or now-previous RSP's keys and our own keys, and transferred it to our distribution server.

This meant that any RRSIGs that were on either of the zones that had been distributed over the last 24 hours would have validated, as all DNS keys involved in those transactions were signed by the KSK that has been validated through the DS record in the root.

Next slide, please. More questions?

UNIDENTIFIED MALE: Be gentle.

ROBERT MARTIN-LEGÉNE: Hi. I think it looks pretty good, what you did. It looks very much like what everybody else is doing in cases like this, especially the part about separating the key changeover from the actual nameserver changeover, which is really two different things, just related to the same process of you having a commercial contract that you [want.]

One thing you didn't write I think was that you removed the name – if you go two or three slides back, when you remove the nameservers of the losing registry, did you do it in their zone?

DAVID PEALLE: Yes. The reason for that is that –

ROBERT MARTIN-LEGÉNE: No, I agree, but it wasn't –

DAVID PEALLE: I can explain that the contract for the servers from the old RSP ended on the day. We couldn't have services from their

nameservers being active on the day of transition, and therefore prior to the day of transition – 10 or 12 days prior – their servers were no longer serving requests for the zone.

A lot of the timing is around the fact that the day of transition was a termination of the contract of service. You can do it many different ways, but that's the way we did it.

ROBERT MARTIN-LEGÉNE: It just wasn't clear what point that you would – where you were removing them from the zone, but you didn't say that it was in the losing registry's database.

DAVID PEALLE: Yes. As I said at the beginning, the cooperation between the –

ROBERT MARTIN-LEGÉNE: Is essential.

DAVID PEALLE: Is essential in this. And they were really great.

ROBERT MARTIN-LEGÉNE: Okay. Good. Thank you.

EBERHARD LISSE: Great. Anything remote? Thank you very much. Give him a hand, please. I think this is in fact a thing that happens quite often, or will happen quite often, and it's good to have a book or it's good to have experiences from other people, how they do that successfully. I'm getting to the stage where I'm finding that I cannot multitask. My wife says this is because I'm not a woman, but I find doing one thing at a time, waiting for it to happen and waiting for what's the consequences before you do the next step is the easiest and the best way if you have time.

Alright, Fred Baker, are you here now? No, he's not. Then we'll just move on with the agenda and see whether we fit him in later. The next one then would be the – Moakofi?

MOAKOFI KAMANGA: Moakofi.

EBERHARD LISSE: I'm so sorry. I don't speak your language, and your name probably means something and I can't know what it means. Please talk a little bit about your setup in Botswana.

MOAKOFI KAMANGA: Thank you, Doctor. Good afternoon, ladies and gentlemen. My name is Moakofi Kamanga from Botswana. I work for Botswana

Communications Regulatory Authority, BOCRA in short, and we manage the .bw ccTLD. I'm going to take you through our journey to DNSSEC deployment, sharing our experience.

Can I move on to the next slide? Okay.

First, I will start with a brief introduction to the .bw registry, take you through our DNSSEC roadmap, all the way to the lessons and challenges that we faced during our deployment.

Next slide, please.

.bw in a nutshell, .bw has been delegated to BOCRA, and we employ the three-R model. The registry is open to international registrars, and anyone and everyone is allowed to register a .bw name, regardless of geographic location. And we use the CoCCA registry software. Currently, we have about 6,461 active names. We delegated the .gov.bw to the Department of Information and Technology to run. It's a government department, and currently .gov.bw is not signed.

Next slide, please.

Our journey with DNSSEC began in 2010, ICANN Nairobi. That's when we came across this name, DNSSEC. At the time, it did not make sense what DNSSEC was. It was hard to understand. So, we started reading, reading, reading, until there was some light at the end of the tunnel. In 2013, we set up a DNSSEC testbed

with some help from our colleagues in Zambia, .zm, just to play around and get a feel of what DNSSEC is, because reading and hands-on are two different things. So, we had to play with BIND and OpenDNSSEC and explore it and see what we can get from there.

In 2014, we hosted the ICANN DNSSEC roadshow, facilitated by our good friend Alain. After he gave us the thumbs up, we decided that we will implement DNSSEC in-house as opposed to outsourcing. In 2015, we published our [DSL] root. This year, in 2017, we our waiting on our DPS, DNSSEC Practice Statement.

Somebody would ask the question, “Why did you not start with the DPS?” I hope some of you will agree with me that DNSSEC is a bit difficult to understand when you start. So, we wanted to understand and know what we’re doing before we can start touching things that we didn't understand.

So, this year, we are going to finalize our DPS, because we do have some changes that we want to make to the current setup. So, after doing all those changes, we will finalize our DPS.

Next slide, please.

So, this is the current setup. It is simple, [inaudible] configuration. We use OpenDNSSEC. We opted to use SoftHSM, but that will change. And then we decided to keep the defaults,

KSK one year and ZSK 90 days. And as you can see, we have two hidden masters, one for signed zones and one for unsigned zones. And I must highlight that the network is IPv6 ready.

Next slide, please.

Yes, this is the interesting slide. We have had some outages, and I want to say when DNSSEC breaks, everyone looks at you. So many questions. You'll get to hear from people that you never knew before, asking a lot of questions.

In December 2015 – that was Christmastime – everybody had gone for holiday. We had a power outage that somehow affected the UPS, and our signer for some reason didn't wake up. It took a long time to resolve this, firstly because when it happened, the key personnel was not around. He was on holiday somewhere. And the second reason, I must admit it was due to inexperience in dealing with a DNSSEC outage.

We had the likes of Cisco coming in, trying to assist. And whoever thinks is good one to assist in matters like DNSSEC, it shows how big the outage was. But finally, yes, we managed to resolve it 6th January 2016.

September 20th again to September 25th, we had another outage. This one was a weird one, because initially, the nameserver address affected, it was a secondary server maintained by

Botswana Telecom, was just working fine. This one was a [special] failure. It affected the .co.bw zone. We had people trying to assist, good colleagues from PowerDNS. I saw [Kumar] somewhere there, he also tried to assist.

But the issue here was that the administrator of the nameserver had put his DNS [version] string as [DBJ DNS,] so everybody thought it was a tiny DNSSEC problem. But it happened that the nameserver was not responding with any DNSSEC data for a [inaudible] on the .co.bw, so all the names under .co.bw turned bogus. How did they resolve it? They updated BIND, and somehow they worked.

November 21st to November 23rd, 2016, we had another failure. This was purely human error, because it was a KSK rollover that went wrong. But, yes, it was resolved.

Next slide, please.

What are the challenges that we are facing? Key management is one of the challenges that we have since picked out. It's also difficult to monitor DNSSEC. We haven't really come across a tool that could really help us to monitor DNSSEC. The other issue is failover. When our signer breaks, everything breaks. We don't have failover, and that's something that we want to look at and rectify.

Next slide, please.

What are the lessons? We learned that DNSSEC is easy to break and hard to fix. I know some will not agree with me. I guess it comes with experience. The most important lesson was never ever try to remove the [TS] at the root when you have a problem, because you'll never get to solve the problem that way. That's why key management is a very important aspect when it comes to DNSSEC. With that, I conclude my presentation. Thank you.

EBERHARD LISSE: I have a question from the chat. I was not there when the slide came up. You used OpenDNSSEC?

MOAKOFI KAMANGA: Yes.

EBERHARD LISSE: With SoftHSM?

MOAKOFI KAMANGA: Yes.

EBERHARD LISSE: Do you think OpenDNSSEC contributed to these issues? Not that I'm trying to beat on OpenDNSSEC, but it was critical this

morning and it has been involved in several failures in the past. And what you're saying, it's easy to break, hard to fix, makes me think OpenDNSSEC has a role.

MOAKOFI KAMANGA: No. I wouldn't say OpenDNSSEC had any role to play here. Like I said, the first outage was caused by a power outage, but somehow, yes, the signer didn't come up for some weird reason. The second one was just another secondary nameserver misbehaving. The third one was purely human error. So, I wouldn't say it was OpenDNSSEC.

EBERHARD LISSE: Jacques?

JACQUES LEATOUR: I notice your mean time to repair is getting better over time, but the objective is not to have any outages. When you had these outages, did you reach out to the community? We have tons of mailing lists. We have [DNS Org,] we have TLD Ops mailing list, we have ICANN Technical mailing list. We have lots of different lists that you can reach out and say, "I need help," and that would make your resolution time probably shorter.

MOAKOFI KAMANGA: Actually, the community reaches us first, because it looks like everybody is actively monitoring what's happening in DNSSEC. Before you know it, somebody will be talking to you trying to address the problem. But we always run to Alain, because we started this with him. And, yes, like you said, there is a lot of support from the community. Like I said, we've had people trying to assist, but the community is kind of reactive. So, in many cases, we don't have even to reach out to them.

JACQUES LEATOUR: So, you didn't have anything to reach out to – no specific action items? Because asking Cisco to fix the setup is the wrong place, right? So I'm just thinking you could have sent an e-mail to the DNSSEC Coordination mailing list, and literally 100 people would have been willing to help.

MOAKOFI KAMANGA: Yes. Like I say, we always have people coming in, trying to assist. Many, many of them. Yes.

EBERHARD LISSE: Alright, thank you very much. The focus of this presentation is of course not the failures, it's of course the success. It's always good to point out what worked and what didn't work so that we can learn from each other, but I think taking on DNSSEC is not an

easy task, and getting it to work is one thing and not breaking it is another.

If you just interact with the system, or as I said if you don't roll your keys regularly – I learned this morning that I should do this once in a while, just not to forget how to do this. Robert?

ROBERT MARTIN-LEGÉNE: Hi. You talked about key management. Can you tell a little bit about how you manage your keys? Is it in an HSM? How many copies do you have, and stuff like that?

MOAKOFI KAMANGA: Yes. I mentioned that at the moment, we're just using SoftHSM. And that's one of the things that we want to change because we realize that with SoftHSM, we had a little bit of a challenge with the key management. So, we are looking to put in HSM as we improve our network.

ROBERT MARTIN-LEGÉNE: I think if you have problems with SoftHSM, you're probably going to have more problems with a hardware HSM. It's pretty difficult. You can make it work, but it requires a lot more effort. I think instead if you have issues with SoftHSM, I think that would be something that would be interesting to bring up on some of

the mailing lists that Jacques was mentioning, because if you have a problem, you are probably not the only one.

There are many people who are trying to understand how everything works of this kind, and if you can – like you now explained what went wrong, if you with the same openness can go and say what your troubles are, I'm sure that there's somebody there, even the implementers of SoftHSM probably would be there to help and address the issues. And I'm not saying you shouldn't do SoftHSM. I think it would be great if you could put in a real hardware HSM, but I don't think it would be easier for you. That's my advice.

MOAKOFI KAMANGA: Thank you.

JAAP AKKERHUIS: I'm not talking for the SoftHSM people, but we have used it actually a lot for testing OpenDNSSEC. For security reasons, it doesn't really add a lot, because it's still in the same machine, it's still the same filesystem, and everybody with root capacity might get in and just change the database if you want to. So, it's only for – it was originally [rated] only for test purposes, but people who seem to use it like it. So, it's actively maintained by the original group in Sweden, and they just brought out a new

version, SoftHSM 2, where they actually have solved quite some problems with the original version 1 design. And so, you might want to get directly in contact [via the org] DNSSEC. Usually, they're looking at that, and they are very happy to find problems and try to find out how it's being used in practice. So, that might help there as well.

EBERHARD LISSE: Okay. Thank you very much. So, again, my question, Fred Baker, are you here?

FRED BAKER: Yes.

EBERHARD LISSE: Oh, excellent. Then can you please come on in front? And we'll have your presentation now.

FRED BAKER: Hi there. So, Mark Andrews asked me to give a talk on EDNS0 and the testing that he did with it. He's got 37 slides for 20 minutes, so I'm going to move quickly, and you've got the slides online afterwards. So, if you have questions about the slides, you can go back to that.

So, okay, going to the next slide, please.

The story here is that ISC was testing its cookies implementation and testing EDNS0 and trying to figure out what there was to know about it, and did a fair amount of experimenting with different sites, different implementations, and basically trying to figure out what there was to know. So, I'm going to walk through a fair amount of data and then may have a couple of recommendations at the end.

Next slide, please.

If you go to these places, you have the opportunity to test your own servers and to read about Mark's recommendations coming out of this, and what he's looking at is an issue that came out of his testing.

Next slide.

So, his testing methodology, he tested a series of queries to various servers with different parameters, trying to test EDNS0 behavior, and discovered that it varied really quite a bit. And so the question was, did they do what they were expected to do? And the answer is, sometimes.

He tested the individual extension mechanism and the mechanisms in combination with that. So, he doesn't show all the combinatorics of that, but to give you an idea.

Next slide, please.

So, you can find the report at the link that he puts up here. For K-root servers using their IPv6 address, he found things mostly worked. For L- and M-root servers, that seemed to work. But L using IPv4 for some reason, he got a "Not Implemented" URI. And why? Who knows, but he did. That's just kind of the beginning of the things that he uncovered there.

Other DNS testing, he looked at A-root and B-root, looked at their IPv6 and their IPv4 addresses with – I'm sorry, next slide – with a variety of different options, and looking at how they responded to the various kinds of servers. And I'm not going to tell you everything he's got in his notes here. He's got a lot of detail. But basically, he found that servers will often drop queries with certain flags.

So, next slide.

The aim of the talk, what he's trying to talk about is the current state of EDNS compliance, and the impact of different extension mechanisms that are used without proactive [6] steps to fix current issues.

Next slide.

Now, when he started out, he found that DNSSEC was reasonably well supported, but that unknown EDNS options and unknown flags were not well supported. And there's defined

behavior when you don't know what you're looking at, and things weren't necessarily doing that. So, things were not necessarily following the RFC in terms of the error conditions.

Next slide.

Now, this is a similar test that was done later, and he is showing that the result was better in some cases and worse in others. So, the five groupings are the nameservers listed in the rootzone, the nameservers for the top and the bottom 100 names in the Alexa top million, and the nameservers for .gov and .au in the Alexa top million. Why .gov? Well, .gov is supposed to support all this, and, gee, let's go check it out. Why .au? Well, Mark lives in Australia, so he went after things that were important to him.

Next slide. Yes, there we go.

The blue columns are EDNS version 0 queries with no options, no flags for the SOA record at the zone's apex. And the gaps here are servers that do not respond to EDNS queries unless DO is set to 1 in the request. So, it's an unspecified behavior, and he's basically looking at the result of unspecified behavior. And it's mostly benign because people don't actually do this, or so he says.

Next slide.

Now, the different colors are color-coded there and they go to root and TLD servers, Alexa top 1,000 servers and bottom 1m000, and so on. You can see that there's really quite a distinction between various systems that he tested. The servers for the bottom 1,000 tend to have a high level of content change which results in a noisy measurement, and you can see the noise in the measurement line there.

Next slide.

And again, this shows the DNS version 0 mishandling. The servers here respond nominally supporting EDNS in that they gave a response to an EDNS request in the first series. So, one would expect them to do the right thing. And this is showing that some servers only respond if certain flags are set or whatever. And the green line shows typical packet losses. Only servers that responded to at least one of the test queries were counted.

Next slide.

The orange columns show DNS key query responses to an EDNS query, with the buffer size set to 512 bytes in an attempt to trigger a truncated UDP response in the server. The responder should have sent a response code of [no error,] and if EDNS is 0, included an OPT record. If the zone is signed, there's a high probability that the record will be truncated, and he's trying to force that to happen. So, the test reports mishandling of

unsupported and unknown query types, and if the zone is signed, the mishandling of truncated responses. So, he feels that levels of misbehavior are unreported because there's no way to force a truncated response.

Next slide.

So now we're showing a test being done over time, and you can see that there are some points in time where it changes fairly dramatically for the bottom 1,000 servers. Not so much for the other ones, but specifically for those.

Next slide.

You can see here how servers misgenerate truncated responses. The error rates will be underreported. And the yellow line shows the number of responses without an OPT record present, which is something that one would expect with EDNS. It's about 1% higher than with other datasets. So, the issue here is figuring out why the response failed. So, we're looking there at malformed responses.

Next slide.

The yellow columns show EDNS aware servers that do not mishandle DO=1 DNSSEC queries.

Next slide.

And again, you can see here that the different servers queries are color coded, and they have different behaviors. That seems to have changed about summer – well, mid-year 2015 in the course of his testing, and gotten worse, so whatever that's about.

And now, next slide.

Once again, we're looking at the breakout of EDNS DO=1 DNSSEC queries mishandling. And the blue line there shows DNSSEC-aware servers that don't set the DNSSEC flag. The yellow line is servers that don't return an EDNS response at all, and so we would expect the NSID option to be present. And packet loss is also slightly higher than DNSSEC responses.

Let's move one more slide.

The green columns are servers which handle unknown EDNS options correctly. This is the good case. Unknown EDNS options are supposed to be ignored by EDNS-aware servers, and we show a fair percentage of the servers in fact doing that.

Next slide.

But that varies, especially with the bottom 1,000 Alexa servers, and the green line is the Alexa .gov servers.

Next slide.

And [inaudible] the yellow line, this is a breakdown of how unknown EDNS options are mishandled. The graph is taken from .gov Alexa top million servers. Most of them, mishandling is improving, with the exception of unknown EDNS options being echoed back to the client.

Next slide.

This gives you an idea of what errors fall in different places. The three marked lines indicate servers that cause DNS resolution failures with BIND today when validating if they also serve a signed zone. BIND treats FORMERR and bad version responses as an indication that the server doesn't support EDNS at all and retries as plain DNS, which is incompatible with getting a DNSSEC response. And similarly, BIND works around servers that do not respond to EDNS queries by sending plain DNS queries. So, once again you see DNS validation failures. So, there are several forms of errors here.

Next slide.

The green line here indicates servers that incorrectly echo unknown EDNS options. Servers that do this are one of the reasons that the EDNS client [subnet] is only supposed to be sent to whitelisted servers. So, this would impact how future EDNS responses or software is done.

Next slide.

The blue columns are correct responses to queries with unknown EDNS present. These are supposed to be ignored, and in these cases, they are. So, the good news there is we're looking at 90% compliance. Where are we with time?

So, next slide.

Now you can see with the blue line, the root and TLD servers effective maybe August or September of 2015 seemed to be pretty much on top of things. The bottom 1,000 Alexa servers are all over the map, and these others, you can see that they're color coded, different sets of people. The point here being that different servers behave differently.

Next slide.

So, unknown flags are mishandled in two ways. The flag is echoed back rather than being ignored by the server, and firewalls block queries with unknown flags. So, firewalls themselves can be a problem. Surprise, surprise. The echoing back of unknown flags means you can't trust the presence of a flag in a response mean anything, and active directory suffers from this right now. And blocking queries within an unknown flag will impact on the DNSSEC validation, as the resolver can't determine what has been blocked or why.

So, next slide, please.

Now, you can see that this slide changes dramatically, or this set of columns changes dramatically, from previous sets. We're looking at different behavior. The red columns are servers that correctly answered EDNS1 queries. Queries with unsupported EDNS versions are supposed to be responded to with an RCODE bad version, and the version field set to the highest EDNS version supported by the server. So, you can see that there were some failures there.

Next slide.

And what you can see here is the percentage of EDNS-aware servers that passed a plain EDNS1 check. Root and TLD servers, we seem to be in pretty good shape. When we get to the Alexa .au servers, Mark's favorite servers, life is not good. And somewhere in between, we've got the Alexa bottom 1,000 and so on. So, you can see that there's quite a wide variation on whether they can pass a plain EDNS1 check.

Next slide.

And this is looking at .gov over time. So, looking at different kind of responses that he got and failure reasons that he got. So, there seems to be a big change, perhaps September, August of

2016. You can see quite a vertical there. And certainly less timeouts, but various things changing at that point.

Next slide.

So, in this particular test or in this chart, he explains what happened during that time. The big vertical is around somebody turning off a firewall, and therefore having the servers themselves respond to invalid EDNS requests, as opposed to the firewall correcting that. They got different responses. They still saw some timeouts and other errors. And then they eventually turned on IPv6 and they did that in four steps, so you can see the four steps there. And they got – let's see, what kind of errors? No output, basically saying, "I don't understand it."

Next slide.

So, with this, those failures got responded to with plain DNS and eventually got sorted out. The timeouts still happened, and they had a format error in some cases.

Next slide.

And there were a set of cases in which they had no data. The purple line represents servers that incorrectly return responses that could be interpreted as no error, no data unless the EDNS version field and the RCODE field are sanity checked. The vendor

there has been informed of the issue and presumably is dealing with it.

Next slide.

These highlighted columns, the green column show a server which is correctly responding to all EDNS extension mechanisms.

And next slide.

You can see once again the TLD servers seem to be doing pretty well, but the bottom Alexa servers are in the 60% range and are pretty noisy. And the Alexa .au servers are not doing even that well.

So, next slide.

Mark, at this point goes into, so, what do we do about this? What do we learn from this? What should we do? To fix the noncompliance, the first thing he recommends is that people please go fix their DNS server implementations. I gave you – and I think I'll give you again – a link where you can go to an ISC server and have it check your implementation for test purposes, and so that's an important issue.

The firewalls seem to be in the way, so please fix the firewalls. Part of this has to do with testing. We need agreed tests on

noncompliance, what is compliant and what is in fact not compliant, and testing for noncompliance.

There's a policy question. One would think this goes without saying, but, gee, wouldn't it be interesting to say if you have a noncompliant implementation, that you shouldn't use it? Please fix it. And policies related to grace periods and that kind of thing.

Next slide, please.

In this, Mark completely repeats the slide. Why? I don't know.

So, next slide.

So, this gives you two links. One will tell you about compliance, and one will allow you to test your servers, so go for it, please. And he's got a posted Internet draft related to the operational issues here. With that, I'm done.

EBERHARD LISSE: Thank you very much. Any questions? Dave, and then Robert, and then we are about done.

[DAVE]: Hi, Fred. It's a long time. So, in 2007, ICANN's Security and Stability Advisory Committee actually ran similar tests for resolvers and for firewalls to see whether EDNS0 was supported, in anticipation of longer messages. We actually have a page that

we put up quite some time ago that seems like we'd be able to work with you to publish the checks that you use. Our checks, I just checked our checks, ten years ago, the [digs] that we were using are no longer legitimate and valid, but we have a list of firewalls that were actually compliant then at that site and a list of resolver implementations that were compliant at that time. So, it sounds like we should probably work with you, and I'll put you in touch with the people who are now on staff support to see if we can resurrect or synchronize our activities.

FRED BAKER: That sounds like a good idea. Mark's address is marka@isc.org. Copy me, I'm fred@isc.org.

[DAVE]: Okay, great.

EBERHARD LISSE: He was on that list of presenters, so you have it.

[DAVE]: Yes, that's perfect.

EBERHARD LISSE: Robert?

ROBERT MARTIN-LEGÉNE: Hi. I have a question about when you say the top 1,000 domains from Alexa. Is it –

FRED BAKER: Half of which seem to be Google, yes.

ROBERT MARTIN-LEGÉNE: Well, that's a question, because how many times do you count the same name? So, if it's a nameserver for, let's say, 100 domains in that list.

FRED BAKER: Like I say, I'm reporting for somebody else, so I'm going to guess. But when he says top 1,000 names, I tend to believe him as opposed to noting that they have the same address, they're going to the same system or at least the same Anycast address.

ROBERT MARTIN-LEGÉNE: And since this has been running for quite a while –

FRED BAKER: Yes, he's been running it over a period of time.

ROBERT MARTIN-LEGÉNE: So, he's trying to look at common denominators in the data, like one certain hosting company or...

FRED BAKER: Yes.

ROBERT MARTIN-LEGÉNE: Okay.

EBERHARD LISSE: Robert, I'll give you his e-mail address and then you can –

ROBERT MARTIN-LEGÉNE: I think I have it.

EBERHARD LISSE: Inquire to your heart's content.

FRED BAKER: Yes.

EBERHARD LISSE: Okay. Thank you very much for presenting a difficult topic off the cuff. Thank you very much. So, the next one is Amreesh Phokeer from AFRINIC, who will talk about their infrastructure.

AMREESH PHOKEER: Hello. Good afternoon, everyone. So, my name is Amreesh from AFRINIC. I work in the R&D division, and today I'm going to talk about our DNSSEC infrastructure, but also tell you how we also migrated from an older signer to a new signer of OpenDNSSEC last year.

Next slide, please.

So, AFRINIC you know already, we are the ones who manage IPv4, IPv6, [and ASN] numbers in Africa. We maintain what we call a WHOIS database, where there we register all the allocation and assignments we have done to our members, and members can also come and register their own assignments also by [location] in that database.

On top of that, we also provide a few services such as RPKI, which is one component of the domain rooting security, Internet rooting registry, and we also sign our [reverse] zones since a few years now. We also provide IPv6 and other types of training, and as I said since last year, we're operating a small lab at AFRINIC.

Next slide, please.

On top of all those, we are also running three separate DNS programs, the first one being the African root server copy. We have six root servers that we have deployed right now, mostly K

and L. We are also supporting the RFC5855, which are the c.in-addr.arpa and the c.ip6.arpa. And finally, we're also providing support to the community through our African DNS Support Program, where we are offering free secondary DNS servers to African ccTLDs. And we have roughly around 30 ccTLDs in our infrastructure.

Next slide, please.

Today, we are going to talk about reverse DNS. As you know, the .arpa zone is broken into two: IP6 for IPv6, in-addr for IPv4. So, when AFRINIC get resources from IANA, so we have DNS servers for those resources. And when we delegate resources to our members, our members have their own DNS servers. And if they want to activate DNSSEC, they have to sign their zones and publish their DS record to [the parent], which is AFRINIC, to build a chain of trust.

Next slide, please.

So, in terms of resources, those are the reverse zones that we manage, IPv6 and IPv4. We have three different IPv6 zones and six different IPv4 zones. And as I said, members who manage their own DNS servers and zones, basically reverse zones, they need to send their DS records to AFRINIC, and this is done through the WHOIS database.

Next slide, please.

So, this is a typical record of a WHOIS domain object where a member would put their authoritative nameservers, and here also they can put their DS record. This is public data, anyone can come and look for this information.

Next slide, please.

Another way to add DS records or nameserver records is to use our portal, MyAFRINIC. The good thing about the portal is that it also checks the validity of your DS records on the fly and it's very easy to upload domain objects.

Next slide.

In terms of policy, our KSK is on 2048 bits RSA, our ZSK, 1024 bits RSA, and our signature, SHA-256. We rollover our ZSK on a monthly basis and our KSK on a yearly basis, and we use the double DS scheme to actually do the key rollover. Signature lifetime is typically 15 days, and we do TTL, as it's normally done in practice. We do not do NSEC3 because we do NSEC. NSEC is good enough, because data on reverse zone are basically public data.

Next slide, please.

So, this is the highlight of the AFRINIC DNSSEC architecture. As I said, the central aspect is the WHOIS database, where members will come and put their domain information. This domain information is dumped hourly into a hidden DNS server. We also have a zone update daemon that will take the domain, do some cleanup, and then send [row] and sign zones to OpenDNSSEC signer, which is actually still currently running SoftHSM. We have plans to move to a hardware HSM. We actually bought one. We are thinking of buying another one. As you know, it's quite expensive and it's quite something on a budget. So, once unsigned zones are signed by the OpenDNSSEC, they are then transferred back to the hidden DNS server and then transferred to our public DNS servers.

Next slide, please.

Right now, only five members are sending DS records to AFRINIC. If we count AFRINIC itself, we are six. Mark is there, he's smiling. He's one of those, we did it. We are around 1,500 members, so adoption is very, very low. If you have ideas of how we can boost that, you're most welcome.

Next slide, please.

We did a signer migration early last year. Why we did that? Because we were facing some scalability issues with OpenDNSSEC Version 1.3. We had very large delays for signing of

zones, and the signer was actually stuck in flush mode for some unknown reason. And obviously, members have started complaining about their zone not being propagated in time. That version of OpenDNSSEC was not fully supporting AXFR in and out, so that's why we decided to move from this one.

Next slide, please.

So, to do this migration, we had a set of guiding principles. First of all, DNSSEC validation should be maintained all the time. So, there should be minimum or no manual editing of signed zones. That's pretty obvious why. Migration of course should be done as quickly as possible. We tried to keep the interaction with parents, in our case it would be the IP6 or in-addr.arpa zones, at minimum, but that's not really an issue because we can actually interact with the parents as many times as we want. And of course, we don't want to change any policy. We want to keep the key sizes [or algorithm] all the same when we're doing the migration.

Next slide, please.

To do this, of course we had to make sure that in the course of the migration, there was no ZSK or KSK rollover, because then we don't want to get into a situation where we have multiple DNS key. And also, we had to make sure that the validity of the signatures is much longer than the TTL of the zone, actually two

or three times bigger. And of course, the old signer and the new signer should both be hidden primaries and not be the public-facing DNS servers. Of course, both signers should be provisioned in the same way. So basically, if we are sending X amount of zones to the old signer, we need to be sending the exact same information to the new signer. And of course, our parents need to accept double DS, because this is the way that we have chosen to do the key rollover.

Next slide, please.

In terms of migration strategies, we have evaluated four different options against a few criterias, the criterias being invalidity window, key manipulation, rollover time and number of interaction with parents, DNS key size, exposure of private keys, which is very important. The first option being we export the existing keys. So, basically, we have to go into the signer, get the private key [material], export that to the new signer and then do the switchover. The second option is a key rollover. The third option is new keys, so basically stopping everything, have new keys, and then put the new signer up there. And the fourth option is existing keys followed by a key rollover. So, obviously, we kicked option 1 and option 3 out, because option 1 means that we are exposing the private keys. Option 3 means that we are going to have an invalidity window, so we didn't want to do that. Option 4 is also exposing the private keys, so we were left

with option 2, basically having two signers running in parallel and then doing the switchover.

Next slide, please.

So, this is what we did. From the new signer, we generated new ZSK and KSK which we sent over to the old signer. And from the old signer, we took the public ZSK and KSK, we sent it to the new signer. Of course, we published the new signer DS record to the parent, and after some time, we made sure that everything was validating as it should be on both the old and the new. And then when we were ready, we just stop the old signer and continue with the new signer in production. After a while, we removed the old DS, and also after waiting for some time, we removed the old KSK and ZSK in the zone files.

Next slide, please.

So, as I said, we used a double DS scheme. So basically, as I said, we had to publish – at some point we had two DS in the parent zone, and this worked perfectly well. Actually, it would be the recommended solution if you want to do a migration.

Next slide.

So, in terms of future work, so the issue of adoption. As I said, we're actually suffering from very, very low adoption. So, we had an idea at AFRINIC. How about if we tried to host a DNSSEC

engine for members? Members can then transfer their zones to AFRINIC, AFRINIC can sign them and then send it over to the member. The member can then publish them in the public DNS server. But of course, [this come an] implication. Would you actually be comfortable in AFRINIC managing your private key materials? And then how can we actually also make sure that there is validation all the time? Then we have to get into agreements such as SLAs, etc. So, this part is tricky. So, it's an idea that we're still thinking about. I think that's my last slide. Yes. Thank you very much.

UNIDENTIFIED MALE: You have to turn off the red one, Eberhard.

EBERHARD LISSE: Thank you very much. Looks like Robert from PCH has got a question.

ROBERT MARTIN-LEGÉNE: So, I don't need to introduce myself? You suggest DNSSEC for members. Is it just for reverse DNS, or is it for any domain they might have in their portfolio?

AMREESH PHOKEER: Our focus is on reverse DNS, because this is our mandate. The mandate is reverse DNS. As I said, we are also providing support for secondary DNS and for ccTLDs, but our main goal would be for reverse DNS.

ROBERT MARTIN-LEGÉNE: So, you haven't discussed it past reverse DNS?

AMREESH PHOKEER: Yes, the issue being scalability and resource, and etc.

ROBERT MARTIN-LEGÉNE: Yes.

EBERHARD LISSE: Any other questions? Okay, thank you very much. That gives us 25 minutes' break. Let's all be here at quarter past to 20 past. The next presentation will start at 20 past.

[END OF TRANSCRIPTION]