ICANN72 | Virtual Annual General Meeting - Tech Day (3 of 3)
Monday, October 25, 2021 – 12:30 to 14:00 PDT

KIMBERLY CARLSON: Hi, everyone. Thanks for being with us for Tech Day, Part 3 of 3. As a reminder, these calls are recorded and follow ICANN's expected standards of behavior.

Comments and questions will be handled via Q&A and the chat pod as they were done in Parts 1 and 2. You're also welcome to raise your hand and make your question or comment verbally.

With that, I'll hand the call back over to you, Eberhard. Thank you.

EBERHARD LISSE: Thank you very much. Next speaker will be Greg Freeman from Lumen, who will talk about the RPKI deployment experience at his company. Can you say a word or two?

GREG FREEMAN: Yeah. Can you hear me okay?

EBERHARD LISSE: Okay. We can hear and see your presentation.

GREG FREEMAN: All right. Perfect. Thank you. So thank you for the time today. And good afternoon, everyone. My name is Greg Freeman from Lumen

Technologies. As was mentioned, I'll be talking a little bit about our experience with RPKI.

Now, as a reminder, RPKI is a security mechanism with works with the BGP stack. And what we wanted to do is just take a few minutes to talk about our experiences, our lessons learned, and some challenges that we had, as we as why we deployed and, more importantly, why we think you should employ it. And we would encourage you to employ it as well if you haven't.

Go to the next slide, please. So Lumen's network is a very vast, expansive network. We have over 450,000 route miles of fiber in the ground. We have over 150,000 buildings that are on-net. And we have over 300,000 IP devices under management, as well as 25 billion minutes per month of voice use on that infrastructure. So it is a very large ecosystem. Any time that we discuss making very large-scale changes to that ecosystem, we have to be very thoughtful and deliberate in our approach.

The network that we were looking to implement and change are the ones in the top right. These are our largest IP networks that we have at Lumen. So Autonomous System 3356 is ranked as the largest autonomous system on the planet. It has 43.82% of all prefixes sitting behind that. And if you're interested in more of those statistics, you can just Google "largest ASNs," and what you'll find is that the [inaudible] Institute is a computer research firm out of San Diego State that analyzed a lot of the prefixes and autonomous systems on the Internet. And so we rely on that a lot to see how we're doing as we're continuing to move around some of our networks from our mergers and acquisitions.

The second-largest network that we have is Autonomous System 3549. That one's ranked 26th globally at 2.5% of all autonomous system sitting behind that. And then CenturyLink, Autonomous System 209, is ranked 36 without 1.69% of all autonomous systems sitting behind it.

So those networks are very, very large in scope, and 3356 is the one we're going to be discussing more in depth here in just a minute.

In addition to that size, Lumen Network has communication services in 60 countries and all continents. And, yes, that does include Antarctica. We've deployed some of our communication services for the research station there in Antarctica.

We also own Autonomous System #1. So if you're looking at who's AS1, well, we're AS1. Today, we have most of that for internal, and then we have those two large networks.

If you think about all of these networks, in the way that the Internet has evolved over the last, really, 20-30 years, it was set up on mutual trust. And all of these networks that we had are typically products of different mergers and acquisitions. You can still 3356 as Level 3. 3549 was Global Crossing, and 209 was CenturyLink. So we've continued to merge these things. That mutual trust … Typically, people would get an IP assigned to that specific ASN inside of the route registry, and then that would be sent over. We developed a number of toolsets and we changed that configuration and the routing ecosystem and updated the prefix list on those routers.

And over time, as all these companies … There's been mergers and acquisitions, and some of the IP address space has been released, or customers come and go. Our optimization of keeping that up to date could be much better.

So what we've been doing is we've been moving away from that open network "trust everyone," and more to a "trust and validate" or "trust and verify" model. And that's where RPKI are coming into the picture.

Next slide, please. So our RPKI is the Resource Public Key Infrastructure. And as I mentioned, it works in conjunction with the BGP protocol stack. And what we're trying to do with it is we want to be able to prevent BGP route hijacking, hijacking of our IP space. So RPKI is this ecosystem that helps validate: is a specific IP address originated by a specific ASN? And based off of that RPKI validation, we can then see if this is something we want to reject from a specific route. Do we want to accept it or do we just want to ignore it? There's a few different options we can take there.

And so with our implementation, what we've done is we wanted to stop and make the Internet a little more secure, specifically around the ones that are misconfigured, maybe the ones that have been left and cybersquatted for decades on it. And realistically we wanted to get rid of the ones that are part of human error.

But the ones that RPKI will not do in a lot of these things is malicious hijacks. If there is a determined attacker, RPKI in its current implementation will not take care of those malicious type of events if it is a determined attacker. But there are a number of use cases that it will provide. So even with that caveat, we still feel that it is beneficial to the broader Internet to adopt and move forward.

So if you look at the diagram a little bit there on the right, what you see is the different route registries will take … And today they will be signing a ROA. And once that ROA is pushed down, what we had to do is we had to turn up a number of what you see called servers. Those are the RPKI servers that we had to deploy across the globe. If you remember back to our network diagram, we're a global company in nature. So we had to put a lot thought into how many servers we wanted to put, what regions we wanted to put, all the routing schemes, [inaudible], and everything that goes with that.

So what we decided to do was treat it almost like a DNS-type of deployment, where we would use Anycast IP addressing for all of our RPKI server infrastructures. We went with a dual vendor technology for our server space. And the theory is, if you look at the ones starting in the E.U., if one of those servers goes offline for another reason, it will just roll to the next Anycasted server that's in the region. And then, if all of the servers go offline for whatever reason, if that's just due to problems or an attack or what-have-you, then it would effectively roll back to the instance where we didn't have RPKI deployed in our network. So we put a lot of thought around resiliency and redundancy in these failure-type scenarios.

The second piece is down at the PE layer. So the PE is to enable RPKI. This is something that we've talked about for well over a decade. But there was an amount of work that we'll talk a little more about on the next slide on the well that we had to move forward. So one was we had to make sure the PEs could handle all this load, and we had to change the configuration of all the PEs to now use the RPKI implementation. So if you think back to the thousands of assets we have that the servers would have to interact with and the PEs would now have to interact with, there was just a lot of coordination that would have to work.

One more thing on the malicious attacks before we move. It was mentioned that this was a BGP stack. So if there are static routes or social engineering or a few other techniques, RPKI will not prevent those. But even with that, we still feel that it's worthwhile.

Next slide, please. So as we started to deploy RPKI, part of the challenge was this is the way it's been for the last … ever—20, 30 years. Why should we deploy it now, and what's the business case justification we would have? And so if we just look at it from a pure financial standpoint, those business cases could always give a little bit of challenge. So what we wanted to do is we wanted to make it more about security of the Internet as a whole and how we want to expand and enhance the customer experience with more reliability in this security stack. So that CX or Customer Experience was always top of mind for us when we were deploying this.

Now, with that, we had to work through the RPKI software flaws. On the server stack, for example, we had to patch the servers multiple times to get the software that we wanted. We test all server and PE ecosystems in the lab before we push anything to a large production network. But even with that, there's certain things you don't catch, unfortunately, until you get it in the network. So we had to work through some of the software on both the PEs and the server side.

Second, hardware requirements. We run a multi-vendor routing network—so Juniper, Cisco, and Nokia. Nokia is one of our big vendor providers as well. And so we had to look at all of the permutations of hardware that we have in the network. Will they be able to handle the RPKI protocol? Is their load considerations. And which ones are so old that we should not attempt this with them? So we did a small amount of [GRUs] in our network to make sure that our ecosystem will continue to work.

Second, we had to work on our software pack. So, as I mentioned, we upgraded the server software pack a couple of times. We also had to, unfortunately, upgrade our PE or Provide Edge routers more than once. We got through the deployment of most of the network, and then we had a software anomaly and we had to do that again.

And what we found with all this ecosystem is that that software-planned activity in the customer communications is really just as important, if not more important, than anything you're doing in an RPKI implementation. The validations servers and the customer comms piece … If you think about it, you have to validate all the work that you're doing on the PE shift to tell customers, "We're going to take your service down for that." And then, once the ROAs start meeting in force, customer routes are no longer going to be accepted in certain instances. You have to get out in front of those customers if you don't want your customer experience to be poor.

And so what we had to do is we had a number of engineers that started looking at all of the routes that we had on … which ones that … as soon as we turned on RPKI, where we would discard invalid entries, who would be impacted. And then we had to get in front of those customers to tell them, "This is the date we're going to enable this, and you're going to have some impacts if you take no action." So working through, with our end customers, all of the iterations and all of the IP changes that they would have to make with the ROAs and education is no small event.

And so, in our communications, there were a couple thigs we could do. If it's valid RPKI signature, we would allow that in. If it was unknown, we would assume good intent and we did not enforce any unknown. But if it was a violation, those are the first phase that we were now enforcing. So educating that customer base is really a key, key point in this as you move through it.

There were a number of other things that we had to look at inside of our RPKI deployment. As I mentioned, we did choose to groom some of our hardware [gear] out of the network prior to even implementing this, but we also wanted to do a lot of cases studies on how much load and how much routing would be increased on all of these various platforms as soon as we turned on RPKI because we didn't want to break a large section of our network and our customers' as well.

The routing policies. I mentioned we used Anycast. Selecting the vendors for those RPKI servers. There's only a handful of them in existence. So selecting a couple of those we thought would make good tool vendor strategy. And making sure they all worked was one of them. And then support and training. Your staff that takes customer calls for the history of the Internet in our industry, in our company, never had to look at RPKI. And so educating the workforce when customers have complaints to make sure they know how to check this, to make sure they know how to look at the ROAs and get all of that working, was a key piece, too, as well as just trusting the undersigned regional route registries.

Next slide. So with all of that said, effectively, rolling out RPKI is a fairly complex undertaking. And at Lumen, it was very complex. We had to take a very careful and deliberate approach to get all of this out. And one of the key benefits beside the enhanced security and that we're now bringing to our network is, with the ROAs being able to correct some of those invalids, there's a lot of IPv4 space that can be reclaimed. If you think about just how much IPv4 space is worth today and how scarce it is becoming, that's a real benefit, both from an economical standpoint and just an optimization of IPv4 usage, that's big in the industry. Customers like that. We like that, too.

But the key punchline, the key thing that I'd like you to come away from, if you don't remember anything about RPKI, is, yes, there is a lot of work and a lot of due diligence that's needed up front, but we believe it's worthwhile. We believe that, if more providers and more network operators will enable our RPKI, we can make the Internet a more secure place and we can move it forward, even past the constraints that we have today.

So I would respectfully ask you, if you haven't deployed RPKI, to please consider it. Together, we can make the Internet a little more safe.

So with that, I wanted to open it up to any Q&A on RPKI and any questions anyone might have.

EBERHARD LISSE: Thank you very much. Very interesting presentation.

Any questions?

I see no hands. There is something in the Q&A. Rubens Kuhl from Brazil asks, "Does Lumen intend to use AS0 RPKI feeds?"

GREG FREEMAN: I'm not aware that we've looked at that at all and intend to at this point. Most of the things that we've looked at this to this point were around our enforcement policy on, "Do we just accept the validates?" which we do. If it's a clear violation, get rid of those. But we still have a very large percentage of unknowns. And so those unknowns at some point we would like to turn on and start enforcing those. But today our thought process is, if we did that, there are much larger sections of the Internet that we would break.

So that's been our strategy. I know it's been the bulk of the discussion. And nothing really on the AS0 there.

EBERHARD LISSE: Another question from [Andy Marliss]. And this is the last question that I'm going to take. "What do you see as the number-one issue"—oh, there is another one. We'll take two. "What do you see as the number-one issue with wider deployment of RPKI?"

GREG FREEMAN: I'll go with two. So one is just enforcing the unknowns. So, if you think back to RPKI, we're looking to see if that origin—that's the key principle—ASN is allowed to propagate that IP. And so with that origin ASN, there are ways to get around that. And if you enforce the unknowns, there's a large percentage of "It's not the origin. I don't see it in the registry. Is that valid? I don't know. It's unknown. We're allowing that to move through." If we enforce that, that'll break a larger piece of it. So that's the number-one thing that I see top of mind for our RPKI.

The other one is we talked a little bit about malicious actors. If someone has [hide] intent, they could likely get around RPKI, and it's because it's only looking, again, at that origin ASN. It's not looking at any of the path ASN. And so the lack of enforcement of unknowns, because the broader Internet hasn't applied, as well as that, are the two things that are top of mind.

EBERHARD LISSE: And then the last question is from [Ben Overander from Rep]. "In your experience, what is missing in the RPKI software ecosystem?"

GREG FREEMAN: So, on the server side for the RPKI validators, there's only a handful of providers on the planet that do that. We started out with three of them, and one of them told us they were getting out of the RPKI validation software business. So that left us really with two. And so what we wanted to do is we wanted to be [Ansible-ize] all the templates we were thinking we would push down, and we wanted to be able to use that on the server stack. And when we deployed those two providers that we selected, I wouldn't say … It was limited selection. The software was okay. We worked with them to patch it. But just that limited number of vendors in that environment is probably the biggest limitation we've got right now.

EBERHARD LISSE: All right. Thank you very much again for a very interesting presentation.

And therefore we are giving the floor to Roy Arends, who will talk a little bit about a hyperlocal root zone service.

ROY ARENDS: Hi, everyone.

EBERHARD LISSE: We can hear you and see your presentation.

ROY ARENDS: Thank you, Eberhard. I'm still Roy Arends. I'm still here at Tech Day. I'm going to discuss hyperlocal with you.

What is a hyperlocal root zone? That's a collective term for using the root zone local to a resolver.

Next slide, please. The hyperlocal term is something you can look up. It relates to a focus on matters concerning a small community or geographical area. And it used to be in the context of local news and weather forecasting, etc., etc., but since the mobile app era, it's now used for weather apps and local maps and local services. It's for the provisioning of data for all of these apps. In that sense, the hyperlocal root zone is basically a resolver that uses the locally available root zone instead of the root servers.

Next slide, please. The concept is not new. It's certainly not invented by ICANN. I think it was Steve Crocker that coined the term "hyperlocal root zone," or "hyperlocal." But the idea of serving a root zone within a resolver or close to a resolver or in the same network as a resolver was suggested by Paul Mockapetris in 2003. It has been suggested by many folks since. It's part of research by David Malone in 2004, "Hints or Slaves." You can find it in many user groups on the Internet in the last ten years and how to do this.

So operators already do this. So we thought it was time for a technical analysis. Why we do think it's time for a technical analysis is because it's part of our root nameserver strategy and implementation. And this is OCTO-016 that was published a year ago plus one day, on the 26th of October, 2020.

Next slide, please. So how does hyperlocal impact a resolver? We are going to look at various aspects and also various viewpoints, including that of the operator, and how to deploy this. So there's query privacy and root zone integrity that we're going to look at, as well as query latency, telemetry, and operational complexity.

First off, query privacy. A lot has been said in recent years about DNS and privacy so this should not come as a surprise. In the context of RFC 6973, DNS servers are observers. And what are observers? Observers are entities that can observe and collect information from communications, potentially posing privacy threats. Now, we know that DNS data is collected passively at observation points. And this is often referred to as passive DNS. We know that DNS data is kept for a long time and distributed to third parties. And we also know there's no transparency on how DNS query data is collected, stored, processed, analyzed, used, shared, or sold. We know there's some efforts to make this transparent, but we also know that there's no clear standards on this.

The good thing is—there's a lot of bad news—that query minimization and aggressive native caching really helps to preserve privacy. So if you're considering something remotely like hyperlocal, please consider query minimization and aggressive negative caching as well.

The main thing here in the context of query privacy is that the hyperlocal root zone deployment avoids the need to send queries to root servers. The query not sent is of a course of query that can't be collected.

Now, when you deploy a root zone—next slide, please. Oh, sorry. I should have said this before. Yeah, I just mentioned this. Sorry. Next slide again, please. Thank you. I'll be more [considerate]. Root zone integrity. Most of the records in the root zone are not DNSSEC-signed. None of the delegation point NS records and glue records have signatures. Now, that is by design. That's not an accident. There's nothing wrong. This is by design.

We also know that there's no transport security between root servers and resolvers. So, in this sense, a hyperlocal root zone provide better integrity than individual response coming from root servers, provided, of course, that the root zone is securely retrieved or securely checked. You can do that clearly with HTTPS, with PGP signatures, or TSIG via local roots. Local roots are something offered by ISI. Wes Hardaker is heavily involved in that. I recommend that you look at it if you're interested in hyperlocal root zone deployments. But in the future, we hope that there's a DNSSEC-validated ZONEMD records.

Now, what's a ZONEMD record? A ZONEMD record contains information about … It's basically a hash of all the records in the root zone. That hash is then DNSSEC-signed by the zone-signing key, which can be trusted via the root zone via the key-signing key, etc., etc., which hopefully you have a trust anchor for.

Next slide, please. Query latency. This is not a big thing. A query to the root zone is often a resolver's first query in a series. It basically doesn't have the information of where—I don't know—let's say, .com or .net is, so it will ask the root servers first, which basically blocks the rest of the series of queries that is coming. But this only happens sporadically, when the information is not available in the cache. So in terms of that kind of query latency, we have to wait for a query to the root servers. These are exceptions. They do happen. We know there's a fair amount of traffic to the root servers. But avoiding root servers, doesn't really lower the latency in that sense.

However, we know that about 68%--and I just checked today on ITHI; it's currently, in the last three months, about 63% of queries to the root zone—return an NXDOMAIN. These queries basically are for domains that do not exist.

Now, we know that Chrome browsers send a large amount of nonce labels, which causes a lot of processing. And these responses will be cached by resolvers and they will cost memory consumption, etc., etc. We also know that root servers spend a lot of time … I mean, if 63% of the queries to the root return a NXDOMAIN, then we know a that root servers spend a lot of time answering these queries. We also know that Google—sorry to point it out—is one of the organizations that sends a lot of queries and is working on a fix.

So a hyperlocal root zone lowers latency, causing better throughout for all queries. It's hardly worth [the slides]. It should be a small motivation here.

Anyway, next slide, please. Reduced telemetry. So we know, for instance, that DITL (Day In The Life), with two projects run by DNS-OARC, recorded 48 hours of DNS traffic to the root servers in some top-level domains. They published that for DNS research. Of course, if root servers do not get traffic or do not get this traffic due to hyperlocal, then you also miss some interesting telemetry data, such as the deployment of new features, v4 versus v6, UDP, TCP, etc., etc., or new EDNS options, but they could be observed elsewhere as well—for instance [inaudible] top-level domains. So the paper that this slide is about actually talks about all of these elements in far more detail than I can do here.

Next slide, please. If you look at elements of deployment specifically, you need to consider the following things: availability, or “Where do I get the root zone from? Do I get it from root zone operators? Do I get it from IANA? Do I get it from a root zone maintainer?” These are not suggestions. These are options that are available. There probably a few more. What kind of transport are we going to use? Is it FTP, HTTPS, AXFR?

The integrity is important. How do I know that the root zone that I’m receiving is correct, is the right one? In the future, I can use ZONEMD plus DNSSEC, but currently I’m either using PGP or TLS. What’s also important is the idea that you need receive timely updates, right? How do I make sure that I use the latest signatures of a root zone DNS record? They don’t last forever. They time out after a certain amount of days. So you need to timely update them. Also, the root zone changes regularly on a daily basis. So, on one hand, Notify is handy, and ISI’s local root offers this, but you should check any if you have the latest.

Then there’s the idea of a fallback mechanism. What do I do, where do I go, when my hyperlocal fetching of a route zone fails? So then the ide is to make sure you use the root hints again or maybe, if not, a fallback mechanism. All of these things are important.

Now, if I look at—next slide, please—the operational complexity of the deploying a hyperlocal root zone now, as in today, there are a few issues there. So current security provisioning is cumbersome. Local roots, luckily, offer TSIG, but as a shared secret, that actually doesn't scale very well. I'm sure it scales nicely at moments, but when you have thousands if not millions of deployments, then TSIG becomes problematic.

Then Internic.net—that's where you can download the root zone from—uses HTTPs. But these TLS certificates are guaranteed by Certicom and not IANA. And maybe to an outsider that's almost the same thing—it doesn't really matter who certifies that—but it does when you talk about the root zone. It needs to be guaranteed by the IANA key-signing key [inaudible].

There is something like PGP that you can use, but that's really cumbersome in an automated environment. I mean, how do you roll the PGP key? Then you have things like local disk management, simple file write rights, and cronjob management if you want to do all these things by hand. And some of these elements are already addressed by modern implementation, but also each implementation, whether it be KNOT or BIND, really has their own method, their own way of doing things.

But what remains an issue is the cryptographic zone file integrity check, and that is until ZONEMD is actually deployed in a root zone.

Next slide, please. If you look at implementations, there are basically three ways, from a 50-mile-high perspective, of doing this. A resolver can serve authoritative data. Now, that's a method that basically everyone recommends against, but technically it's possible. The problem with that is that, with an individual client's implementation, they may not see the AD bits on root zone content because it comes with an AA bit. It comes directly from an authority server. LocalRoot ships this configuration. I'm not sure if they still do it. A resolver can also use a local authoritative server for the root zone. You can either do it on a network or a loopback or use an internal mirror zone. RFC 8806 has this configuration, and BIND uses the term "mirror zone" for this configuration.

Another way of doing things, which is really interesting, is … Sorry. I lost my [train] for a second. Oh yeah, it's priming the cache with the root zone itself. KNOT does this. What KNOT does is [inaudible] resolver from [CZDNIC]. Knot basically fetches the root zone. It checks it using its certificate—a TLS, I think—and then it basically primes the root zone in its resolver cache, basically just like you prime it with root hints. Now, the nice thing about this is that it times out. And when it primes out, you basically reprime this. You do it once a day. So it's a very nice little deployment. And I think that's also mentioned in RFC 8806.

So, in conclusion—this is my last slide … Next slide, please. Sorry. In conclusion, a hyperlocal root zone is not new, has been deployed for years. Current software, recent software, makes a hyperlocal deployment easier. There are benefits such as better integrity and better privacy and a slightly better latency. There are also some drawbacks, such as less telemetry and observation points and additional operational complexity. We know there's still some work to be done in regard to ZONEMD. We also need to make the root zone maybe more available or using a pool or root zone publishers. That's all food for thought.

You can find the full paper on this link that I published. It's the last bullet point on this slide. And that's it from my side. Any questions? I'll hand it over to Eberhard. Thank you.

EBERHARD LISSE: Thank you very much. Also quite interesting. Any—I see a hand there hanging on. There's a speaker from PowerDNS. Can we please enable … Can you please unmute yourself? You have the floor?

UNIDENTIFIED MALE: Hello. Today, we released an alpha of a PowerDNS recurser with cache priming quite inspired by what Knot does from [inaudible] or a website or a local file. Thank you.

ROY ARENDS: Fantastic. Well done.

EBERHARD LISSE: Thank you. Any other questions?

Peter, could you take your hand down, please?

And then Jaap Akkerhuis mentioned in the chat that Unbound does local root for well for at least a year.

Good. I see no more questions. Thank you. Sorry. It's the Q&A. Is there some—Vladimir Cunat from Knot Resolver, "Is there some timeline for getting ZONEMD published in the root?"

ROY ARENDS: I know that the RZERC has issued a paper for the ICANN Board to consider. I think this question should actually go to IANA because I honestly have no idea of an actual timeline on getting ZONEMD in the root zone. This is something that you go to either IANA or the RZERC folks for. Maybe one of them is on the call right now. Thanks.

EBERHARD LISSE: Duane Wessels?

DUANE WESSELS: Hi. Can you hear me?

EBERHARD LISSE: Yes.

DUANE WESSELS: So for information on ZONEMD getting published in the root, I suggest people try to attend the DNSSEC and Security Workshop in a couple days where there's going to be a presentation on that.

EBERHARD LISSE: Thank you very much. That looks like no more questions. So we move over to the last presentation but not the least one. Dmitry Kohmanyuk from .ua will talk about do-it-yourself of the ccTLD infrastructure. You have the floor. Can you say a few words? We can tell you whether we can hear you.

I can't hear you yet. I can't see you in my panelist. We seem to have lost him. No, he is in the Attendance. Can you promote him, please? There you go.

DMITRY KOHMANYUK: Right. That should do it.

EBERHARD LISSE: I can hear you. Thank you. You can go ahead.

DMITRY KOHMANYUK: All right. Having redundant home connectivity is a must nowadays. I had to order my secondary fiber link to the home, and it looks like that's okay.

This talk is more a bit of a meditation looking back at our now 20-year history of writing the independent ccTLD registry. So I hope it would be a bit useful to other folks. And of course, I tried to fit more [ideas] that can possibly fit into that slide deck, so please, next page. And I'll try to mention things.

So imagine that you have to run a ccTLD and you just got the contract. You're a new CEO. Maybe your country has just formed. Let's see what makes out that [successful] operation that you can possibly do yourself or maybe you're doing some of it, or all of it.

Next slide, please. So these are the main parts, I guess. And I'm just mentioning that as the UA operator since 1992 as an independent collective and since 2001 as a company. So of course, as DNS service for the TLD or TLDs you run and maybe, I would say, probably your own domains because I would never trust any other party to run our own section domains like our own company, maybe some other zones. You need DNSSEC-signing and key management structure and your zone generation, which basically means that all of the database records that you take from your registry and create a zone file to distribute. WHOIS and nowadays RDAP services. And by the way, we do run RDAP in .UA, but it's not in full production yet. Your websites for public, for your registrars, maybe other partners and maybe users if you deal with registrants directly like I know DENIC does. We actually only do this for our government and educational customers because those are non-commercial zones. EPP service. Now I guess it's something we should have to do, although there may be registries that don't do that yet. We used to have a web interface, but we basically don't do this anymore. And then have your usual office support lines, e-mail, maybe fax—hopefully not.

Next slide, please. And these are maybe less visible but of course important things. Datacenter space. And those are in the actual [inaudible]. Your Internet connectivity, your networking, hardware and firewalls, routers, switches and whatnot. And then of course the software has to come from somewhere. So maybe you're doing everything outsourced but most likely aren't. And in our case, we had most of our stuff written in-house with exceptions noted [a few slides down.]. Then your system network operations, devops, if you're doing it in a new way or maybe you run old-school stuff and actually have mostly stable software [just updated on a monthly or maybe less cycle.] Then your business office and stuff.

[inaudible] specifically, I believe most of the trouble that comes with running ccTLDS, apart from keeping up with Internet standards and customers keeping happy in a privacy and policy sense is actually responding to various legal issues. They may come from government, from your registrars, most likely from disputes. We have implemented UDRP and I would tell you this was like night and day. We aren't subject to many litigations, but of course we have our share. I'm very happy and I would recommend to anybody. But, yeah, you still need a significant, let's say, consulting capability, maybe external companies. I don't think anybody wants to hire a lawyer as a staff person. Maybe a paralegal. We don't.

Next slide, please. And those are the choices that we have made and that any other registries would have made. And again, as I was looking back at 20 years of our development, I would not say anybody should do the same as we did because nobody is in the same shoes. I guess even ten years ago we had done things differently.

Next slide, please. And those are things that we have done. And of course, nowadays doing your own DNS maybe only makes sense for the primary server, although we do run our own Anycast. I would say probably you don't want to run your own EPP and WHOIS service unless you outsource the whole thing. And that's my Point #8. So, in a way, in my honest opinion, either you found a good partner like—I'm not going to name any companies there, but you know them—that would do it for you, or you would do most of the stuff yourself and maybe just delegate commodity things.

In my opinion, most of the things you are responsible for are either custom-made or something that is only available from a small variety of vendors. So, for example, I would not probably outsource my registry database software or [inaudible] because those are pretty simple database structures, but we probably should outsource our financial accounting software now.

I would consider those as put into the "must keep." And "maybe not" is the registry data, whatever you put into [escrow] database, if you have any, and your primary financial information, like your customer data, your private customer—I mean, registrars being the customer, or maybe direct customers if you're dealing with them—and, of course, financial stuff, your DNSSEC infrastructure, maybe you run certificates already. We do. It's not on the slides, but we do run our own certificates already for the purpose of our operations.

I would say the presentation on DNS core had an influence on me. I clearly see how small the core of DNS operation worldwide is. It's, what, a few thousand AS numbers? It's actually a small amount. It's better to not create a circle of dependencies, as the most recent cases of Facebook have shown us. Doing it yourself is fine, but you should be prepared for the worst.

So let's do the next slide, please. And here are the other components that I would not recommend anybody actually build, except maybe, as an exercise. Of course, you're not going to write your own DNS server, although I do know people who've done that in the past. I'm not talking about software vendors but some of our registrars who did that. Mailing lists, accounting software, database. We use [inaudible], we use Proxmox and stuff like that.

But again, if you're doing those things and you are diving into details and you want to optimize either for money or time, I guess the extremes of do-it-yourself and getting a company that can do it all for you both make sense. I think trying to migrate your [existing] hosted system into, well, may be partially outsourced. I would say Cloud-hosted [inaudible] data makes sense. But once you go past core functions, I think doing the rest yourself doesn't make sense. And the core functions would be registry and financials.

Next slide, please. So I guess costs here mean that you're either optimizing for the short-term or long-term costs or likewise for the capital operating expenses. In our case, we have built most of our infrastructure in [2000] and have been consistently revising it and writing necessary components. But even EPP software we have is our. Our DNS management is our own, our DNSSEC-signing [inaudible] our own. But whatever was possible to get ready for, we did. But I would not imagine doing things that you are required as part of your maybe legal requirements or country legislation off the shelf.

I know maybe two or three companies in the European region which are doing something akin to the full hosted registry. One of them is called [Registry IO.] They're from Bulgaria. I'm not saying his solution is the best. It's just one of the examples. I know CommunityDNS used to do something like that. I know you can have non-EU companies. I specifically mentioned the European Union but let's say the European region companies. But I think, again, you're going to the point where customer requirements and the amount of time to implement them into something ready-made will probably exceed of the cost of just doing it yourself, especially now when the software costs being actually lower than before.

Next slide, please. So, again, maybe it's a strong point, I would say. It was probably a bit challenging to write software itself in the '90s, but now you can get custom development cheaper, and components and Cloud services and toolkits are more available. So I would probably say doing this with strategy in mind is probably better than just trusting somebody else to do what's right for you, as no ccTLDs are specific to IDN tables, local presence rules, whatever customs we have. For example, in the Ukraine, we had a challenge of supporting IDN names and supporting parallel secondary level domains which have variants of Ukrainian-Russian-English translation of city names. Those things like that—the deeper you go, the more sense it makes to implement your business logic [inaudible]. When it comes to commodity, DNS servers, sure, go ahead and just get it. PowerDNS or Bind or Knot DNS all work well.

Let's go to the next slide, please. I think I'll get some acknowledgements. Well, that was not meant to be all-inclusive, but I would say I was very grateful to folks from various [strata] of our little DNS worldwide operation and [inaudible] operation group.

I will just mention—please turn over to the next slide—several groups, of course, company employees, and partners we work with, our ccNSO and general TLD community. I guess ccNSO now is almost the same because enough countries have joined. The staff of ICANN and staff of RIPE-NCC and of course DNS-OARC and the RIPE community and CENTR members.

Again, I say we had very long road, and I would not try to include an entire list of things that are making a good ccTLD. But I would say doing this on your own is, at the same time, easier and harder than before. And to make a short pun, you're [the final authority] of what things are making of the task in hand. We've been blessed with having no big government regulation, but I know other folks did.

So with that, I guess I will ask for some questions. And because it's the very last talk and we're probably all tired, I guess I would not bore you with more details. My original plan was to give you more details about our infrastructure, but I just think it would not be appropriate for such a relatively short talk size. So I'm looking forward to expanding this maybe to [inaudible] meeting or another forum. Thanks.

EBERHARD LISSE: Thank you very much. I always like to have—

DMITRY KOHMANYUK: Thank you, Eberhard and [inaudible].

EBERHARD LISSE: You're welcome. Thank you. But I always like to have presentations like this, a little bit about infrastructure, a little bit about how people operate, best practices, in the sense not that we're bound by them, but documentation things, business continuity, because that's what this Tech Day has always been about and should always be, and not only for ccTLDs but also for small gTLDs. I find that many gTLDs fail, especially smaller ones, because they tender out everything out and think this [inaudible] money. It's not really that easy.

Anyway, I see a hand in the attendance from Andrey You have the floor, please. Unmute yourself.

ANDREY SHCHERBOVICH: I'd like to pose a question to Dmitry. I'd like to attract your attention about the [inaudible] of, for example, the Russian government to create the sovereign Internet and Alternative DNS system in Russia, actually. How should we be sure, maybe by technical means, that it's not possible to create systems like that, or if these systems only make damage or harm to everyone? Because we need to ensure the integrity of the Internet actually. And technical means also should be … Because this alternative Internet and sovereign Internet will not be the Internet. It'll look like the North Korea network. I'm not a technical expert. I'm a policy person. Sorry for this question, but [inaudible]. Thank you.

EBERHARD LISSE: Dmitry?

Dmitry, have you got an opinion on this?

Okay. He has been unpanelized. Can you make a panelist out of him again, please?

KIMBERLY CARLSON: Yeah. We just moved him.

EBERHARD LISSE: Thank you. Dmitry, you have the floor if you wish to answer.

DMITRY KOHMANYUK: Yeah, sure. I'm sorry, looks like it died just when I was waiting for the question, so I guess I haven't got the first part of it. I was [inaudible] since the mid-sentence. I apologize about that.

EBERHARD LISSE: He basically—

DMITRY KOHMANYUK: I'm looking at the Q&A but I don't see the textual part of it—

EBERHARD LISSE: He basically wanted to know what your opinion is about "sovereign" Internet and Alternative DNS in Russia.

DMITRY KOHMANYUK: Well, of course, the Ukraine has all the authoritative opinion about Russia. I think it's the silliest thing you can do.

EBERHARD LISSE: Yeah.

DMITRY KOHMANYUK: The quantity of the units of the network quadratically influences its usability. So basically, quadrupling the size makes it 16 times more useful. I don't know. Those are approximate numbers.

And also, I guess it would have worked in the '90s. Nowadays I'm really doubting it. I know what the Russian government is doing, but all of my friends and connections in Russia are not using Russian websites or Russian resources only. They use global resources.

So, yeah, you can make do with [inaudible] your own country if you're blessed with a good climate. You can probably do with e-mail and chat and web from your own country. If it's Russia or China or even the United States, would that make you satisfied? I doubt it. But I guess it's a bit beyond the DNS operation and ccTLD operation in particular.

That's more my own opinion. I think it's an experiment which result just proves that the idea is bad.

EBERHARD LISSE: I mean, you all know my view that I think governments should keep out of this.

But, anyway, Vadim Mikhailov asks, "Do you consider usage of any registry backup systems?"

DMITRY KOHMANYUK: Yeah, we do. I guess the biggest question to that is, do you trust the third parties to do this for you? Because in our case, we just have off-site as well as on-site replicas of the data. And our opinion is that, in the case of something really bad happening, we can just restore from the cold backup. The thing that would be a problem is that we'll probably not be able to keep continuous operation when some disaster strikes. But that's a small thing because it's more important to have operations of the data as being updated by your last registry update.

But if you mean backup system, like a warm or hot or cold backup, I guess it depends. If you do outsource your registry, then those people would have it. If you don't, then you should build it. But, yes, I think you should have, at least at the minimum, a copy of your system ready to be used. It doesn't have to be an automatic switchover, but you should have a procedure and disaster training, disaster preparedness training, which you would use in case you do fail on the primary [set].

Anyway, this is where the Cloud makes sense because running that one in the Cloud would probably be a good idea because you basically would be saving on continuous costs of maintaining the secondary site, yet you have something to fall back to while your primary server infrastructure is recovering. I'd probably not advocate being the Cloud first or the Cloud only—

EBERHARD LISSE: I agree. It also depends on the size. For example, as more a ccTLD, we don't have a hot standby, but we are so warm that we would probably not even lose a renewal. And we have turned out our registry system today for 24 hours because we're doing a backup restore exercise tomorrow and I don't want registrations to interfere with this. And tomorrow we'll do an [upgrade.]

The point is, if the registration system goes down for a day or two, that's not a problem as long as you don't lose any registrations or renewals that happened in the meantime before you noticed or between the last backup and between the failure. That's a problem. But if the registration system is down for a day, then you can't register, you can't renew, and you can't delete, and you can't transfer. Tough luck.

DMITRY KOHMANYUK: Oh, sure. Yeah, that's actually what we need to take care [inaudible] EPP, we made automatic renewals. So basically, in the case of Ukrainian TLDs, all domains are renewed forever until explicitly cancelled. That has a nice side effect. In the case of the registrar going bankrupt or missing in action or having their own disaster, the customers are not going to have an impact. What would happen is that whoever of their customers would have wanted to [inaudible] their domains would actually have to pay for another year. And I guess that cost would probably be compensated by the failing registrar. Or in our case, the registry would say, "Those were gracious renewals. We just don't charge you for them." We haven't done the latter, but we did do the former. Like when one of our registrar went bankrupt, we just said, "Look, we give them a month extra to move their domains to somewhere else. And during that month, whoever got expired got a free year+ because that's our policy—that the registry should never drop the domain unless explicitly requested—

EBERHARD LISSE: We have a registrar of last resort, and one of our registrars failed, we moved the domains over there. We don't remove them. We don't delete them. We let them expire. And that usually takes care of it.

Anyway, thank you very much, everybody, and particularly Dmitry for coming up on short notice with this.

DMITRY KOHMANYUK: Thank you.

EBERHARD LISSE: So I'm going to hand the floor over to Jacques to close Tech Day.

JACQUES LATOUR: Thank you, Eberhard. So as usual, in our tradition, we do a closing remark/quick assessment of the day. So the first presentation was from Chi-Yuan on IoT device identity in a 5G context. That was interesting. A lot of people are trying to figure that space out. So it was nice to see an option in there.

Ed Lewis did a presentation on DNS Core Census. It was interesting. It's almost 4,000 zones in the core DNS. So looking forward to seeing the data sets if they're made public. But Ed is looking for expressions of interest. So if you're interested in this topic, let him know. The e-mail address is at the end of the presentation.

After that, we had a presentation from Garth on CoCCA, the panopticon Domain Security Initiative. That sounds a whole lot like one of those Transformer robot things. It was interesting in that I think it's a framework to fight abuse for ccTLDs. I like the approach. I'm looking forward to seeing the next presentation to see what's going to come out of it. The framework looks pretty cool with RDAP and users having system access [and] RDAP registration data to do its abuse fighting work. So I'm really looking forward to seeing the next version of this.

Eduardo did a presentation after that, which is the RDAP conformance tool. So hopefully one day it'll be an open-source tool that we can use to do our RDAP conformance testing. There's over 30 test groups and about 200 tests to look at.

DNS magnitude from Roy. That was real interesting. I followed the work that Alexander did way back on DNS magnitude, but this is a use of that tool and public reporting of that data. Until today, I didn't know about the observatory.research.ICANN.org. And there's magnitude content there, reports from the root. I was happy to see there's data in there. Dot-CA has a magnitude of 8.303. So that's good to know.

Ed did another presentation on DNS algorithm choices, issues around SHA-1 usage and issues around DNSSEC algorithm rollover. So lots of stats around that.

Viktor did a presentation on DNSSEC parameters for TLDs. I think that's the best practice for implementing DNSSEC in TLDs. There's an analysis of where we stand. There's lots of crypto stuff with KSKs and DSKey, algorithm signing, NSEC. Today we do have issues with RSA 1024-bit keys for ZSKs. So people that have older ones could be at risk. So they need to address that sooner than later. We need to upgrade either to 1280 bits or to do an algorithm rollover to ECDSA. So this was noted by a few people: this should be a regular presentation and to show us where we stand on that.

We had a RPKI presentation on deployment experience from Lumen. I had to Google it. Level 3 is now Lumen. I didn't know that. Now I know. It was a nice presentation in understanding how to [inaudible] the high-level issues around implementing RPKI, the lesson learned in the deployment, the challenges, the issues. So overall the work is a good benefit for the Internet, but I think, as a community, we have a lot of work to make it more robust.

Good presentation on the hyperlocal root zone service. So Roy is planning all of it, all about the privacy, integrity, and more stuff. Interesting enough, it seems like a super simple idea or process to do, but there's a lot of things you need to consider before you implement. So it's all in there.

And Dmitry did his short presentation on best practice to build your own ccTLD and the things to consider. So it's all in there.

And that concludes my remarks for ICANN Tech Day 72.

EBERHARD LISSE: Thank you very much, everybody. We'll see each other, I hope, in person in Puerto Rico. Goodbye.

JACQUES LATOUR: That would be so awesome.

[END OF TRANSCRIPTION]