ICANN72 | Virtual Annual General Meeting – Tech Day (2 of 3)
Monday, October 25, 2021 – 10:30 to 12:00 PDT

BRENDA BREWER: I hope you all had a good break. Welcome to Tech Day Part 2 of 3. As a reminder, calls are recorded and follow ICANN's Expected Standards of Behavior. Comments and questions will be handled via Q&A and chat pods as they were in Part 1. You are also welcome to raise your hand and ask your question or make your comment verbally.

Eberhard, over to you to kick things off.

EBERHARD LISSE: Thank you very much. For those who haven't been here before, there are two ways of doing this. You raise your hand and you will be recognized at the end of the presentation, or you ask your question in the Q&A pod and it will be recognized at the end of the presentation.

Garth, you have the floor.

GARTH MILLER: Thank you, Eberhard.

EBERHARD LISSE: And we can hear you and see the presentation nicely.

GARTH MILLER: Okay, perfect. Thank you very much. A little bit about the background of the project. Historically, within the CoCCA registry system, we used to connect to the Secure Domain Foundation API which was a nonprofit in Canada, just to check on malicious activity. And they aggregated data from a variety of places, and it sort of gave us a free, central place to actually check for malicious domains and that sort of thing.

That API hasn't been available for a while, so we looked at some commercial options. We found a few that were very nice. But for many of the small users of the CoCCA software, small ccTLDs paying for a commercial service to monitor abuse was not something they were particularly interested in. So we decided to essentially duplicated much of the effort of the Secure Domain Foundation, but catered more to ccTLDs.

One of the developers that used to work for the SDF has worked for [inaudible] for about three years. So he was sort of familiar with what was required. That's essentially what the initiative is about, and it's a set of tools. It's a free service, free software—depending which way you want to go—that can be used.

Currently we're just looking at ccTLDs and smaller ccTLDs because there is quite a bit of effort in collecting the data and validating the data. We're not doing UK or China or any of the really large ones. But we do look at data on pretty much all ccTLDs from the data sources that we're able to access. Some data sources are free. Some of them are subscription based. Some are commercial. And we're looking at changing that. So if we can jump to the next slide, please.

So as part of the project, we made some basic assumptions. DNS abuse and mitigation is trending. Just based on our experience, a lot of what is considered malicious or abusive domains are actually not necessarily malicious registrations but compromised hosting or domains that are created by registrants that are not registered with the registry.

So at the registry level, I think most modern registry systems have tools to actually block the obvious typos of PayPal or whatever or DGA domains. So at the registry level, if one desires to, it's a fairly straightforward exercise to block obviously malicious registrations. But in our experience, most of the domains that are used maliciously—or many of them, anyway—are actually created by registrants at a lower level, at a subordinate domain level.

So that's pretty tricky to actually block at the point of registration. So what we were looking at is ways, essentially, to notify registrants directly about possible compromised hosting or violations of the policy. So if we can jump to the next slide.

So we made some assumptions, particularly with small registrars in a lot of the ccTLDs, that they have very limited capacity or incentive to actually monitor abuse related to the domains and their portfolios. And a lot of the existing tools that we looked it, commercial tools that we were using for Christmas Island, for example, was very nice. It had a way to automate sending messages to the registrars, but it doesn't necessarily reach the domain contact. And what the registrar does with it once they get a notice, is it something they consider actionable or not?

So it's sort of, given the nature of abuse, we thought it was better to design a system where the registry could actually contact the domain contacts directly. We can jump to the next slide, please.

So our assumption there is basically that the domain contacts, if they want to keep their name working and resolving, they have the highest incentive to actually remedy issues. If it is a malicious registration or a domain that's being used maliciously—free URLs, free hosting, and those that are used for phishing and that sort of thing—it's beneficial to actually put the registrant on notice, essentially, that whatever they're up to with subordinate domains is being monitored.

The other issues that we assume, and not everybody likes to do this, but the registrar—the commercial or the policy environment of the ccTLD—allows for direct contact of domain holders which are essentially clients of the registrars by the registry. So that's our base assumption.

So our goal is essentially to develop tools which collect and aggregate data, and then have a tool to actually notify the domain holders directly instead of going through the registrar. So it's basically bypassing notifications, bypassing registrars and sending notices via e-mail directly to domain contacts. Next slide, please.

So the objectives, as I said, to provide ccTLD managers with current and verified data. So it's basically a daily report of everything that appeared in any of the feeds that we were able to validate or the results of our validation.

And to actually have a tool using RDAP which allows … A useful things of RDAP is that with credentialed access you can see redacted information. So one of the issues with the secure domain foundation when all of the WHOIS servers started being blocked through the GDPR, a lot of their data harvesting activity was essentially not particularly useful. And since the WHOIS servers all have different output, you'd have to parse …

So we wanted a common standard way to actually access public data in the registry and also redated information if we have credentials.

And the idea was, basically, that he tool doesn't require zone files. And it doesn't matter what registry system you're running. It basically works outside of whatever registry platform you want to use or are using. And that, yeah, it doesn't require any development, doesn't require any access to databases or zone files or any of the registry infrastructure. Next, please.

Again, key features. Does not utilize zone files. It's GDPR compliant in that the credentialed access to the RDAP server to extract the e-mail addresses in order to send address to domain holders, that component this actually run by the TLD manager themselves. It's not part of the larger database. If a ccTLD is not using RDAP currently, we've tested with Red Dog, which is what we're using internally at CoCCA, and found that to do whatever we need it to do. So we're quite happy with that. Next slide, maybe.

So there are two components to the project, or the initiative if you will. One is essentially the data collection and validation. So that's sort of

run by CoCCA. Essentially we connect to all the different feeds. We go through a bunch of steps, which I'll go through shortly, to try and validate the information that we collect for the feeds. But it doesn't have any information that's personal or confidential in the central databased.

And that data is basically collected and then, for every ccTLD that wants to participate, we would send an e-mail with a link to the results of the data collection effort on a daily basis. And that e-mail would have a PDF report and also all of the CSV and JSON files and everything related to the actual data that was collected and validated so they get all the detailed data that's contained in the report.

And they also have a small program which is currently just a RAR files which they can then configure a little JSON locally and run the file once a day and it will connect their RDAP server, collect the information for the domains that have been flagged, and send the e-mail from their e-mail server, their e-mail address. That way, really, there's no data that's actually leaking outside of their organization.

So essentially we deliver the data to them daily and they have a small app that we provide that allows them to fill in the gaps in the data with private information and send the reports to the domain contacts. Next, please.

Just a little bit about the process. So we basically collect, I think, about 28 different sources now. I think four of them are commercial or subscription based. The rest are free sources. There's a lot of duplication in the sources and there's a lot of normalization and

mucking around with the data. You might have the same URI from one source with the backslash and not another. Some of them might have Port 80 and another one might have Port 443. So there's basically just a daily effort to go through and collect all the data and then [inaudible] and clean it all up so that we have a clean data set.

Then we're using the Public Suffix List which we override with a small JSON file, because it's not accurate in all cases, to distinguish between domains that are registered in the registry and subordinate domains created by domain contacts.

And then we walk through and for every domain, once we've isolated the actual domain component of it, we check the TLD servers to see if it's delegated or not. A lot of the data from the feeds may not be accurate or up to date. If we do have RDAP access, we would have the EPP status—server or client hold—and we would use that as well to analyze whether the domain is actually active.

So we're trying to come up with a list of actually validated-on-a-daily-basis threats. So we're not particularly interested in history over time or tracking things. We're basically looking at, as of today, what are the current domains being maliciously used or abused, whatever terminology you care to you.

So we check the TLD servers to see if the domain is delegated. We check PCH's Quad9 to see if the domain has been blocked by Quad9. You know, they have their own data sources and they have their commercial arrangements for that. So it just gives us a secondary check to see if one

of the domains that's actively delegated is being blocked by a filtered DNS service. Next slide, please.

Then we use a commercial service. We use Bright Data which is quite a nice proxy thing. So we look at each individual URI and try and decide whether or not its active. So we're looking at HTTP status for each complaint and we look from two different …

You know, hackers might sometimes have a different response depending on where the query is coming from, so we use randomized IP addresses and we look from two different sources. So in that little JSON file when we [configure] ccTLD, we specify to actually check using a data center or IP addresses in that country. And then also we would check from outside of that country.

And then we go through … Google has an API, so we query the Google API to see if the information in the feed has actually been flagged by Google. We use a Tranco list as well, which is a list of popular top-level domains, just to get some indication of the risk posed by that particular domain as far as, is it a variant of a popular domain but just registered in that ccTLD. Next one.

Then we collect information through RDAP. Is RDAP is enabled—the zones are enabled—we can crunch some additional data, and then basically we provide the data. And that can be run locally. Next, please.

So again, why RDAP? Because it gives us a common API that we can connect and have credentialed and non-credentialed information,

redacted information. And it allows us to do what we would like to do but remain GDPR compliant. So that's … One more. Last one, I guess.

I think I'm just repeating myself here, but we're GDPR compliant because the contact information is collected locally and sent locally by the TLD manager not by CoCCA. Next slide.

Essentially, yeah, the TLD manager fills in the blanks from the CoCCA data set and can send notices to registrants from their infrastructure.

Last one. I think that's it. So I'll take questions if anybody has any.

EBERHARD LISSE: Thank you very much. Very well-presented presentation, I must say. And interesting to collate all these things. I'll take it offline with you how to modify it if necessary to approach the registrars. We would not deal with end clients. But I'll take this offline ahis is probably not interesting.

Rubens Kuhl asks, "Are malicious URLs tested with different user agents? We have some actors in Brazil that only show the [phish] for mobile users."

GARTH MILLER: Yeah. Interestingly, we can certainly … If there are known issues, with the proxy network that we're using you can select "ISP data center or "mobile networks." It's a commercial service, so we have to pay to use it. It costs about $30 a run per day at the moment.

And they have a system where it will try connecting through the data center, then through an ISP. And if you set it up that way, then it can test through mobile networks. So that's certainly an opinion to do the testing that way. But it's an additional expense, so testing through the mobile networks through the proxy system is more expensive than testing, obviously, through data center IPs.

EBERHARD LISSE: I'll take one more question from Jacques Latour.

JACQUES LATOUR: Hi, thanks. Nice presentation. So the question I had is, I think I missed it, is this a service offered by CoCCA?

GARTH MILLER: Yeah, it's a service. And we've written the software to essentially collect all the data internally, but we'll provide it to any ccTLD manager that's interested in it. We'll provide the daily reports and also can give you the small software that you can run locally to just fill in the blanks of our data set with your internal data.

It probably doesn't make sense for ccTLD managers to do all the collection. It actually takes us about 10 hours a day to do the run because some of the feeds are commercial feeds. So if you're a small ccTLD, you're probably not going to want to pay the feeds to join the phishing group or get the data from Malwarebytes or something. So it's

designed as, we would provide you the data and then a small app. You have an app that you run locally.

But certainly if you're interested in doing the collection and everything yourself and you have the resources to pay for the subscriptions and everything and you want to do it at scale, then we're open to that as well.

JACQUES LATOUR: All right, thank you.

EBERHARD LISSE: That also answers a question from the chat. And Viktor Dukhovni asked, "Would notification of persistently bogus signed delegations be in scope for this project?"

GARTH MILLER: Can you repeat the question?

EBERHARD LISSE: It's in the chat. It's the last thing in the chat. "Would notification of persistently bogus signed delegations be in scope for this project?"

GARTH MILLER: Yeah. It's a fairly new project. To be honest, we've been working on it for about three months only. So we're open to any suggestions on how to improve it or [inaudible].

EBERHARD LISSE: Okay. I don't want to take anymore questions, Viktor. The e-mail address is on your agenda e-mail, so you can communicate with Garth that way.

All right. Thank you very much, Garth. Very nicely done, I must say, even on relatively short notice. Thank you again.

And the next speaker will be Eduardo Alvarez, and he will talk about the RDAP Conformance tool. You have the floor.

EDUARDO ALVAREZ: Thank you. Can you hear me?

EBERHARD LISSE: Yes, we can.

EDUARDO ALVAREZ: Perfect. Thank you very much. Good morning or evening or afternoon to everyone. I'm going to present this stand-alone tool that has been developed working with a contractor by ICANN. So let's begin. Next slide, please.

So the ICANN RDAP Conformance tool that we developed, as I was saying, is a stand-alone. It's an open-source tool. And the objective is to verify RDAP servers to make sure that they're implementing the specifications developed in the IETF, the RFCs that define the RDAP

standard, and optionally to verify some of the requirements that are defined in the gTLD RDAP profile.

Some key details—this is an application that is meant to be available for free. It's developed in Java 11. It's going to be a command-line stand-alone tool and it will support a flexible configuration so that, depending on each policy or requirement by registry operator or registrar—or whoever operates this RDAP service—can enable or disable specific checks as needed.

These checks or tests, we define them in different groups. As we can see in the table in the slide, we have 27 test groups based off RDAP standards or the IETF RFCs. And these are comprised of 212 tests. Then the gTLD RDAP profile will have 11 groups of tests that include 74 individual tests.

Each of these test groups, for example, are just a categorization of groups. For example, tests that are related to a specific object class such as a domain or name server that's related to different structures that are part of RDAP responses such as a "notices and remarks" array, "entities" array, the RDAP Conformance structure, and so on. So that's why we have this grouping. Next slide, please.

To try to make this a little bit more clear, as I was saying, these tests are basically atomic checks that the unit will be doing based on each requirement defined in the standards. These are defined including a numeric error code. And they would specify the value that's noncompliant and a message describing the issue.

So here we have an example. If we take the test group here that we see on the slide, which is the standard RDAP Conformance validation which is one specific element that can be found in RDAP response, then we have a list of tests. We can see the first one here which basically just checks that "The RDAP Conformance data structure must be a syntactically valid JSON array."

If, when running the tool against an RDAP server that provides a response that does not include a syntactically valid JSON array, then we will see as a result this numeric error code which is 10500. And then the value which is the actual structure that triggered the issue or that it's not compliant. And then a message indicating basically the issue.

And then we will have subsequent testing of this group. We may have from one to many tests in this specific group, depending on what we're checking. But this is basically how the tool categorizes or classifies the tests that it does. Next slide, please.

Currently this tool supports these five RDAP query types. You can use it to verify domain lookup which would be the most common for domain registries and registrars.

You can also check into the nameserver lookup responses, nameserver search, and the help query. Next slide, please.

The basic workflow for using this tool once you download the executable jar or build it from the source code, you as a user just provide a configuration file which, basically—we'll see an example in this next slide—but it's a configuration file where you define which tests

are going to be checked or ignored or classified as a warning only. You provide execution parameters to basically indicate if you want to validate as a registrar or maybe redefine connection timeouts or other parameters that can be optionally specified, and the RDAP URI that's going to be verified.

Then the conformance tool will retrieve some public data sets from IANA. There are some checks that include verifying data from, for example, the IPv6 or IPv4 Special Purpose Address Registry, obviously the bootstrap file to check if the RDAP server [has] a registry, if it's registered or not. And some other lists of public registries that are published by IANA.

Then it will perform the RDAP queries then run all of the validations as configured or specified by the users from either the RFCs that define the RDAP standards, and optionally if the user specifies the option, then also check against the requirements that are defining the RDAP profile. It could be the response profile or the Technical Implementation Guidance.

And then as the last step, as a result, the tool will generate a text file which is formatted in JSON. We'll see a little bit more in the next slide. Next, please.

So for the user input, as I was saying, a configuration definition file is required. This is a standard template where you as a user can just indicate which tests are to be considered as an error which is the default behavior. Or you can, for example, some tests, based on registry or registrar policy or whoever is the RDAP service operator might not be

something that … Or it's something that you're working on or you know is an exception, so you can mark this as a warning or to ignore this specific test. So you can do that in the configuration file.

By providing execution parameters when running the test tool, you can specify if you want to validate the checks that are from the gTLD RDAP profile. And then also you have to specify if you're running … Because the RDAP profile is defined for gTLD registries and gTLD registrars [with] different requirements each, you have to specify which approach you want the tool to take in the verification.

You can specify if it's a "thin" registry, for example, like .com. And you can also provide the option to use locally persisted datasets which are these IANA registries. You can either have the tool get them directly from the IANA-known URLs or just use the URL files that up already have in your file system.

And then lastly, the RDAP URI which is the one that's just for testing. Next, please.

The output file. So here we have a little bit more detail. As I was saying, it's a JSON formatted text file and it's basically a results report which will indicate the URI that was tested, the date where it was tested up, the HTTP status code that was received whether it was a 200 or a 400-something or some other HTTP status.

And then it will have a group, arrays, that will list the test groups that pass verifications with not issues. And then separately, it may have test groups that have either warnings or errors.

And then each of these arrays will include JSON objects that will list all of the issues that were found by listing the test numeric error code, the value that triggered the error, an informative message. And then if the configuration file indicates special notes provided by the user, then those will be included as well. And we can see an example in the next slide. Next, please.

So these are very brief examples how a result report might look after checking a test URI. So in this example—apologies for the small font, hopefully it's readable—we can see at the top that we have a test date, we have an array that is cut short just for the sake of space. But we have an array [that says] "groupOK". And then here we'll list all of the test groups that ran without running into any issues.

Then we have a definition identifier. This is just informative. It will just reflect the comments from the configuration file to say, for example, "This is a configuration file for a gTLD registry" or something else. Or "this is my ccTLD-specific configuration file for verifying RDAP." Something like that. Just a reference comment for users to know when looking at these reports.

And then we have another array that's called "groupErrorWarning". So here we'll see all of the test groups that actually had at least one or more issues detected by the tool.

And then we have the "results" object which will indicate how many warning we saw, how many tests were not verified because they were set in the configuration file to be ignored which is not checked. And then how many errors.

And then we have in red an example of an error. For example, in this case we see an "eventAction" object that included an extra element that's not part of the standard. So it will be flagged as such. There's a specific numeric error code, a message that says what the issue is—just a rough description—and then the value.

You will be able to see the path within the JSON structure, like it's from the top level, then it's in the nameservers array, then it's the first element. And then we'll see the "events" element which is another array. And then the first element of the array will include this "extra" element that is not part of this structure. So this is one example of how an error could be reported by the tool.

And then at the end, we just have "testedURI" and the HTTP status code that the tool received. Next slide, please.

So just by running some quick tests on RDAP servers that we've seen using only the bootstrap for TLDs in IANA, because we don't really have much more to work with. But we do have all of the gTLDs there and quite a few ccTLDs as well. So we have seen that the tool still picks up on some issue that are not uncommon to see. And hopefully having this tool will help implementers of RDAP services to sort of identify issues or just understand if there's something that might be implemented not quite in conformance with the standards.

So some of the examples that we've seen are—and this has come up in other conferences and events as well—issues with vCard array. That's a common problem that we see in many RDAP service implementations. So there are some JSON names that are used that are incorrect. There

are required elements that are missing. For example, in the vCard array the full name of the contact. There is some confusion about how redaction is supposed to be made. So some people just omit these kinds of elements, but this is not conformant with the standard. There are different ways to do redaction, and just omitting these required elements just causing an issue. Or nonconformance with this definition of the standard.

And just syntax issues in values as well. For example, if you say a data element such as the telephone, if included, is a type of URI and then the type is something else, then that could also be reported by our tool, which is also a common occurrence.

We see quite a bit of instances where some structures like the ones that are here—type, status, eventAction—these are meant to use value from known lists that are registered in IANA, and we see some implementations that just define their own types, their own status, or their own type of events that are not registered. So that's something that the tool will pick up and just flag as the issue. "Hey, this is a value that is not registered and it's supposed to be."

Quite a few issues with capitalization in JSON names. Just a reminder that in RDAP and [explicitly] listed in the standards values, JSON names are case-sensitive. So we often see uppercase at the beginning when it shouldn't have been. It might seem kind of trivial, but part of standard is just not conformance. So that's also some of the types of issues that will be flagged and that we see currently quite frequently. Next slide, please.

I'm going to just go real quickly. There are other issues in the RDAP Conformance array that lists supported RDAP extensions. We also see some values there that are not registered with IANA as extensions. That's something that will be flagged, as well, by the tool.

Some issues with the standard "remarks/notices" arrays. There are some elements that are required [in which] every remark, every notice should have their title, their description. In some instances they just have one or the other, and that's not entirely correct for the standard.

Some HTTP headers as well, particularly the Access-Control-Allow-Origin header. That's the one that will allow other web clients to connect to the RDAP service without having issues with the browser where the CORS header—I think that's what it's called in the standard, the Cross-Origin [Resource] Sharing—to be allowed.

And also unrecognized elements as well. Here it's particularly [at least] around the dsData element, but we've seen this in other structures. For example, there are events that have more elements that are not part of the standard event object. They're in the dsData [where also have statuses] or other stuff that's not necessarily part of this data structure. It could be part of an extension but it's not registered, so therefore it's not checked.

So those are the types of things that will be flagged by the tool unless the user specifies to ignore these in the configuration. Next slide, please.

So that's where we are right now and what the tool does. In terms of next steps, this is currently not yet available. It's going to be made available through ICANN's GitHub repositories, but it's not quite there yet. The tool is ready to be used. We're just finalizing some details in terms of the disclaimers, license, language to use. But it should be available soon. We'll probably announced that through an external blog post and probably through our gTLD-Tech mailing list.

There are also other steps that will need to be considered. There are new extensions, new drafts that continue to evolve. We definitely want to add support for JSContact. That's an ongoing work that's happening right now in the REGEXT Working Group in the IETF, but it's currently not in scope. But it's something that we definitely want to get in there at some point.

We know there are going to be newer versions of the gTLD RDAP profile, so there are going to be updates to the checks that we have today as well. That's still currently a little bit more into the future.

And then the other RDAP extensions and RFC updates, we definitely want to incorporate those as they become official, or as they become formal RFC updates. Next slide.

Yep, so that's it. I would close by saying that we welcome all feedback and questions. I'm pretty sure it's going to be more relevant once we make the tool available. In the meantime I can ask to just keep an eye out on the mailing list. We'll probably send an announcement once it's publicly available. And I'll take questions.

EBERHARD LISSE: Thank you very much. You finished right on the dot. Any questions? I see a hand here in the attendance from Brett. If I'm not mistaken, he is from Nominet. Please unmute yourself, and you have the floor.

BRETT CARR: Yes, good evening. Thank you, Eberhard. Eduardo, thank you for your presentation. It was very interesting. I've got two very quick questions. Are ICANN using this tool or planning to do so when they're doing registry systems testing for gTLDs? And are ICANN using this tool or planning to do so within the ICANN MoSAPI gTLD monitoring system?

EDUARDO ALVAREZ: Well, for MoSAPI, no. We're not going to be using this tool because MoSAPI … For those that don't know, the SLA Monitoring System that focuses on response time, service available, there are other purposes to that monitoring, not necessarily compliance with the content of the response.

In terms of registry system testing, at this point this tool is not being used. It may be used in the future. Once it becomes available, I guess that option could be considered. There are plans to make this tool available also to our Compliance Team, just for reference. Currently when we—"we" as in the Compliance function—receive a report of issues with RDAP services, this could be a good resource for reference. I wouldn't call this a compliance check, but it could be supporting

information. So that's definitely one of the uses that could be done to this too. I hope that addresses your question.

BRETT CARR: Yes, thank you.

EBERHARD LISSE: There is one remark by Rick Wilhelm in the Q&A which says, "Can I suggest keeping track of RDAP [minus] redacted for your roadmap pointing towards the data tracker IETF document?"

EDUARDO ALVAREZ: Yes, absolutely. Thanks, Rick. So that's part of the standards and RFC updates that we want to keep an eye out … As soon as new extensions get implemented in RDAP service, the tool might see these as unexpected elements or stuff that is not compliant. So we definitely want to keep an eye out and keep track on this work on newer extensions and updates, particularly those in the regext. Rest assured, we are paying attention and we want to keep those in our backlog to make sure we consider them.

EBERHARD LISSE: Excellent. Thank you very much. Interesting presentation. Also well presented. Thank you very much again. And now Roy Arends will tell us about DNS Magnitude.

ROY ARENDS: All right.

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

ROY ARENDS: Brilliant. I'm Roy Arends. I'm the principle research scientist at ICANN. This is about DNS Magnitude. It's a term you might have heard of before. It's not something we have invented at ICANN. Can I have the first slide, please? Thank you.

So first, of the term "magnitude," what does it relate to? Magnitude basically, if you look at the Oxford English Dictionary, relates to the great size or importance of something or the degree to which something is large or important. And the idea is basically to come up with a ranking that shows impact or some kind of classification that shows impacts.

The term and methodology that I'll discuss here was invented by Alexander Mayrhofer and his team at nic.at. "AT" is Austria, of course. And he presented this first at ICANN58 in March 2017. And I actually personal really liked the idea. It's a simple popularity measure for domains. It's human-friendly, practical, and it's based on actual DNS traffic measurements. And it's mimicked after seismic magnitude scales like he Richter scale or earthquake science. And the scale that we're using is logarithmic.

And instead of counting the number of queries that come in for a domain, we count the number of hosts that query for a domain or the number of resolvers or the number of distinct IP addresses or the number of IP prefixes. And so not necessarily the number of queries for a domain. Next slide, please.

In short, the three principles of DNS Magnitude—I stole this slide from Alexander—we count unique hosts, use a logarithmic scale, and normalize the results. Next slide, please.

I'll give you an example. First, why are we using a logarithmic scale? If you use a linear scale then basically the top ten most popular top-level domains basically dwarf, basically, everyone else. So linear scale is not that handy if you want to see the long-term.

I'll give you a calculation example. I know Brett is here on Zoom as well, so I'll give you an example for .uk which is run by Nominet. If I count the number of unique hosts that ask for something under .uk that we observe in IMRS—that's the L.root server—[and] IMRS traffic.

Then we can see a about 380,000 distinct hosts. If you take the natural log of it, you get a number which is 12.849. And that number in itself is not that interesting. You need to compare it to all the hosts that you've observed over that day. And we take a UTC day, 24 hours, and the overall number of hosts that we've observed is 1,358,000.

So you take the natural log of that and you come a number 14.122. You divided one by the other and you get 0.9. And we basically multiply that

by 10 and you get a magnitude score of 9—9.099 to be precise. You'll see that number later on in some of the tables I'll show you.

So this magnitude in itself is new, but simply using unique sources is not new. And I'll go into that in the next slide. Next slide, please.

So DNS Magnitude at ICANN in 2013. I'm not talking about ICANN Organization, but the ICANN community in general. In 2013 the Interisle report came out. If you remember it was related to Name Collisions and it spoke of a number of distinct sources. And what they did is count IP address prefixes.

Also in 2017, I happened to do a study which relates to the prevalence of DNS queries for .corp, .home, and .mail. And instead of only looking at the number of queries coming in for .corp, .home, and .mail, I also looked at the number of unique prefixes. So we've already started doing things in that realm.

And the idea was basically a high magnitude value or a high number of prefixes that would ask for a domain is directly related to a high collision risk, of course only for domains that haven't been delegated yet. All right, next slide, please.

So having done that, being involved in Name Collisions—not causing them but studying the—and work on the Interisle report that I did afterwards and on the OCTO-007 study. I worked with nic.at, with Alexander Mayrhofer and his team. I provided them IMRS data [under contracts] and they did a study for us. The study is published in the link that you see here. I put the link in the last slide as well.

And that's a really, really good report. It talks, for instance, about how to prevent against gaming the system or abusing the system of DNS Magnitudes and how strong it is and what you need to do. It looks at Country Code Top-Level Domains and domains that do not exist, etc.

So I took their research and I built a tool, a simple website that show the daily top 2,000 ranking of top-level domains based on their DNS Magnitude ranking. And I do that in exactly the way I have described before with that logarithm stuff.

This is observed from IMRS traffic which is basically L-root traffic. ICANN runs one of the 13 name servers. I then aggregated by /24 for IPv4 addresses and /48 for IPv6 addresses. And then I calculate those by a UTC day, so 25 … The reason I say UTC day is because if we don't specify a specific time frame you might wonder if it's related to time zones. It's not. It's just the UTC day.

I also built in a six-day delay. The reason for this is threefold. It allows the measurements to catch up as a maintenance on the measurement system. Sometimes new patches need to be deployed or [inaudible] updates, etc. If it would be real time, we have a harder time to get exact and correct measurements. So that's why we have the six-day delay.

Additionally, IMRS is a system that's fairly fluid. It's a whole bunch of authoritative name servers. And the team that runs those changes these regularly. Things get added, things get removed, etc. And we need to have those changes reflected in the system as well.

And an additional reason for delaying this is that the system can be gamed. The idea is that if I try to influence these statistics now, basically I would like to see direct, immediate results. And if the result is only available after six days, that's basically a [slight demotivation]. Not completely impossible, but maybe it helps us to [demotivate]. All right, next slide, please.

So this is what you see on the web page that I listed before. This is top-level domains tanks by magnitude score. When I say top-level domains, I mean domains observed from queries in the DNS through IMRS. They might or might not be delegated. What you see here is that the [bulk] is actually delegated. There's one special use. That's .local.

And what you can also see here is that the first 10 ranks are fairly stable over time, but I will go through the individual columns with you fairly quick.

The first column is the magnitude score. The second column is the top-level domain itself. It's not ever observed label. They are only the valid labels. So it needs to start with a letter. It needs to be at least two characters long. If there are more than two, the middle can be a dash or a letter or a number. And it can end with a number of a letter. These are the RFC 1034 or 1035 rules for a label.

The third column is the status. The domain can be either delegated or not. If it's not delegated, it's either special-use or not special-use. So it's delegated special-use or the cell will be empty. There's nothing there. You'll see that in a minute.

The fourth column is the ranking—4th, 5th, 6th and 7th. It's basically ranking for that day. The following column is a ranking over that week. The ranking over that month is not exactly a month because months differ in days per month. So I took basically 30 days counting from today's rank, including this day and including this week. And the quarterly rank is not exactly quarterly. It's 90 days and also starting from today counting back.

The coverage. And now you can see where the difference is if you use logarithmic scale or not. The coverage is basically how many sources of the total amount of sources have queried for that name. So it's [essentially] 62% for .com. That means of all the observed hosts that sent queries, 62% sent to .com string. They asked for a domain under .com.

And if you go down this table, you quickly get to 11% and it very quickly gets to … I mean, within the top 100, you get to a 2% or 1%. And like I said, it's fairly low the lower you go. A low number doesn't mean insignificance. We'll get to that later as well.

Then the following column is the unique number of sources that you see. The second-to-last column is the query volume, the number of queries we observed for that top-level domain. And here you can see that there's a different order. For the first five it goes down, but then all of a sudden you have .arpa which is much higher than .uk, for instance.

And then the last column is the average query per source. There's nothing else in the query [but] query volume divided by unique number of sources.

So just looking at the time. I think I'm doing okay. If I sort this slightly differently, can I go to the next slide, please?

So this is ranked by query volume and not number of unique sources. But if you do this by query volume, you see that the table looks fairly different. In the top 10, you now see domains that haven't been delegated like .home and .internal. .local is a special-use domain. It's very popular if you look at query volume. You have .dhcp in there. It's basically all over the place. It's also less stable. If you look at sources, it's much more stable over time than if you look at query volume. Next slide, please.

What the size can do, you can basically click away or obscure or make invisible domains that exist. So you can just look at those that are not delegated. Then you basically get this list.

Many of you will recognized these strings. .onion is here, which is a special-use domain. .invalid is a special-use domain. And a whole bunch more. .workgroup is in there. next slide, please.

This is the same but then sorted by volume. You see .home and .corp in there. .mail is not in this list. It's ranked slightly lower. But already in this list, if you look at query volume alone and then you look at coverage, you can see that almost the bottom half of the page is mostly 0%. And that's not a rounding error. It just rounds down to 0%. That's how low the amount of unique [sources] are. Next slide, please.

Something interesting we've observed is that if you look at newly-observed top-level domains, or newly-observed strains that basically

pop up today that we haven't seen—or at least doesn't show up in the top 2,000 the weeks before, or even months or the quarter before—you see that .xow, the green triangle that you see is basically, if it goes up or down compared to the week before—and you're just witnessing a small bug that has already been fixed today. It should point up, of course. The triangle should point up because it now appears in the top 2,000.

And the second table on the slide is the newly-observed top-level domain for this week, so that we haven't seen in the last months but only seen this week. This is interesting if you look at new deployments of [inaudible]. All of a sudden this string that pops up or software that has been deployed. It comes from somewhere. Someone somewhere has configured these strings, so it's interesting just to keep track of those. Next slide, please.

Actually, this is the second-to-last slide. Now I've sorted them by average queries per source. The sole reason I'm doing this is because you can see the high [inaudible] like .dhcp and .bbrouter. Most of these are configuration strings in networking hardware, if you will. I don't think any of these strings have been configured as a marketing [spiel] to get a more popular, if that makes any sense.

We see the .dhcp in there. .bbrouter, .cmcc, .sercomm, .home, etc. .rac2v1a. You can look on Google and see what these relate to. And some of them, like .openstacklocal, you can see them literally in confirmation documents. All right, so all of this information is available. Next slide, please.

Here you see a list of references of links that I've used in my presentation. And the last link, observatory.research.icann.org/magnitude shows the place that I've just discussed. So I hope I'm on time, barely. I'm going to give the floor back to Eberhard. Thanks.

EBERHARD LISSE: Not only are you on time, you are exactly on time and you have got still five minutes to spare. Juan Antonio Gutiérrez has got a question. Please make him unmute himself. Kim, can you permit him to unmute? There you go. Unmute yourself, please. Juan, we cannot hear you. You must unmute yourself. Okay, not a problem. If you can't unmute yourself, then we can't your question.

Thank you very much for this presentation. I like research. What different—We could have maybe put you closer to Ed Lewis's presentation but it didn't fit timewise. So thank you very much for bearing with us.

Next presenter—

BRENDA BREWER: Eberhard, there is one question in the Q&A if you want to take that one.

EBERHARD LISSE: Oh, okay. Sorry. Justin Mack from MarkMonitor, "How does QNAME minimization affect the governing of these statistics?"

ROY ARENDS: That's a very good question. Not the gathering itself. I mean, it doesn't—the gathering will happen per se, but the ranking won't be that much influenced because we are only looking at top-level domains. So if you have QNAME minimization …

And let explain to some of the folks what that means. It's basically a resolver trick to not divulge too much information. It's a privacy thing. So instead of asking for bot.example.com, you basically ask the root service for .com—the .com service for example.com—and the example.com service for bot.example.com.

In the end, we will still see the .com query, so that counts towards the number of unique sources. However, it might influence query volume a little bit. And we don't know how much, but we do see a slight different between those resolvers to do query minimizations and those that do not. And this might also be related to cache optimization like aggression negative caching. All of these technologies will influence the query volume. They won't necessarily influence the number of unique sources. Thanks.

EBERHARD LISSE: Thank you very much. Next will be Ed Lewis with the second presentation. You have the floor.

ED LEWIS: Alright, thanks.

EBERHARD LISSE: We can see and hear you.

EDWARD LEWIS: Oh, good. Okay. So DNSSEC Algorithm Choices. This is meant to be kind of a lead-in talk to Viktor's coming up next—to kind of coincidentally lead in—where there has been some concern about what cryptography is being used in TLDs. So I'm going to do a short little talk here to cover some of the things here.

So this is a look at the DNSSEC security algorithms used by top-level domains over time. And first we'll talk about what makes this interesting. And then the reason why I perked up when I saw this topic on the Call for Papers is some of the regional differences out there.

So the DNSSEC Security Algorithm is a field in the DNSSEC record that combines two things: cryptography which is the algorithm that's going to be used to perform the signing but also has a hash element. And the hash is used to take the data that you want to sign and crunch it down to a smaller piece of something. And then you run it through the cryptography when you get the signature. That's basically how these two things work.

In some cases you can change one. And in fact, in the old days RSA was the cryptography we would use and you had different hash sizes that were available. The newer elliptic curved ones seem to have it built in together, so it's not as [many] options now as there use to be.

But this was the registry and what I want to show you here is the colors. I have the yellow and the black, and then colors down the side. The reason for our concern here today is to look at algorithms #5 and #7. These are the two that [run]… They use SHA-1 is the hash and they use the RSA signing algorithm. And there's some concern over this.

I'm not going to have many comments on that because I'm not a cryptographer, but there are folks who are highlighting the use of these and trying to encourage the use of the more recently defined DNSSEC Security Algorithms.

Now 5 and 7. They're two different numbers, but they're the same hash and cryptography. The reason for that is a historical reason. When 5 was defined, we didn't have NSEC3. When we defined NSEC3, we had to say, "How do you know NSEC3 is in play? We'll us 7 for the same algorithm."

So 5 and 7 pretty much are the same thing. But because they're of interest in my charts, you'll see them highlighted as yellow and black.

Now digging a little bit deeper here, why are we concerned about this? Cryptography is mysterious. I've never understood it. Many TLDs operators just see it as a parameter in DNSSEC. A lot of times you choose the algorithm base to be just the default that your signing tool will use. You really don't know what's a good cryptography algorithm unless you really are into cryptography.

Now later on, as with everything else in the world, we have technical refresh periods where we want to change things. And sometimes cryptography algorithms need to be changed. They go out of fashion.

They go out of play. They wear down. They get broken and so on. So you have to change them.

Now in DNSSEC is possible to change it, but it's not trivial. So it's kind of a major deal for a TLDs to make a change here. So a look over time, this is going back about 10 years of data, these are the algorithms that have been used. And you can see the large gray area. That's RSA SHA-256 which is pretty much dominant right now.

The RSA SHA-1, the black and the yellow, you see was significant early on and then it kind of lost favor over time. And towards the end of this chart, it seems to crash a bit.

One thing I'll add, too, is that a lot of that crashing happens after the pandemic hit everybody and shut everybody down. So it seems like the engineers went home and did some homework.

Now there's a spikey peak up on top here. That spikey peak there is interesting. That's a sign for not more zones being out there, not more people signing. But in order to change from algorithm to the other, you have to add the new algorithm in for some time, let it get deployed because of DNS caching. And then pull the old one out. So a lot of times you see a spike before a crash, and that's [what we're going to] concentrate on a little bit early here because it's significant.

Before we get to that, though, I have on top here, this is the chart showing all of the TLDs together. Below that I have split this by TLDs on the left which looks almost the same and ccTLDs on the right. And as I

talked about earlier, the TLDs behave differently, and this is one [instance] where this division actually makes a lot of sense to look at.

So for the rest of the talk, I'm going to talk about ccTLDs because that's where all the action is. That's organic growth over time. The gTLDs had to roll out DNSSEC early on, so they pretty much … And they also dominate numerically, so the pictures look the same for them.

So again, the ccTLDs is the same chart. And again, you see this little spike here. It's not a little bit more pronounced. It's actually more of a rectangle than a spike. And if you …

I looked at that for a bit and it rang my bell because I knew that there is a ccTLD operator out there that runs a whole bunch of TLDs, it turns out for the same jurisdiction because in this one jurisdiction there are many written scripts. So they have lots of IDN ccTLDs. And to see what's causing this, it shows that in early March, they added 16 new algorithms. The same 16 zones and signed them in two ways. They let it burn in for some time and then they took it out. They went back to only one over time.

So in the charts you'll see a lot of these spikes. Spikes are rises because there's about to be a change. And the more pronounced the spike, the bigger the operators—operator—[the bigger] the change.

Now this is probably a little deep for a 10-minute talk, but this is another look at the keys that were involved for this particular operator, for one of the operators, rather—or one of the TLDs, excuse me. You'll see here where they have an overlaps of the blue rectangle. And here's green.

This is where the RSA SHA-1, the old algorithm, is replaced by the new one, and there was an overlap because that's the way we had to have it work because of DNSSEC. You can imagine that time when there are duplicate records out there, sizes are a bit bigger than each other. So we try not to do that for too long.

So let me go into the regional because I don't want to take too much more of Viktor's time here. North America, we only have three TLDs that fit the ICANN North American region that are signed. And I'm not going to get … You can probably guess who they are, but early on two of them were signed with RSA SHA-1. The third one came with RSA SHA-256 about a couple of year after. And here's a spike because they were change. A spike here, and changing. And to scale, each of these is one TLDs so it's pretty blocky.

In Africa we see that the chart looks like this. We have more blockiness because we only have about 55 total TLDs in Africa. About 20 of them have signed so far. And again, RSA SHA-256 is dominant.

If you look down here during what I call the pandemic era, the RSA-SHA-1 is holding steady, but it looks like [inaudible] changed to an elliptic curve when you see the first time the elliptic curve appears in [charts].

Latin America/Caribbean region. For the longest time, very stable. Not much growth in DNSSEC. That was 24. Right now we're at probably about 20 out of 37 in that region.

But recently there was a growth. Basically there were additions, but then RSA SHA-256 took over. There were a lot of spikes here showing where it falls back down. And so to this point, we only have maybe one that's still doing RSA SHA-1 algorithms down here and a bit more of the elliptic curve popping up.

In Asia, Australia, and the Pacific Islands, again a lot of steadiness here. There's growth over time. It's getting bigger up here. But now with RSA SHA-1 and SHA-1-N.

Within the pandemic era of course, again people seem to be [factoring] their networks. They're taking it out. And you see that the ECDSA algorithm is starting to pop up here, but still it's only like 3 out of the total. And there are at least 76 there. I can't see because I have Zoom on top of my chart there. But there are quite a few zones in the Australia, Asia, and Pacific area.

Now the last region is Europe, and Europe looks a little bit different. One thing I've found fascinating is that in Europe, the elliptic curve has really taken off. Even before the pandemic. The pandemic began somewhere around here. A lot of folks went acknowledge and say, "Let's change to something new." It's squeezing out the RSA SHA-1, so it looks like SHA-1 is almost out in the European region now. The RSA SHA-256 is still dominant and working, and people seem to be happy with that, with the sprinkling of RSA SHA-512 across the top all the way around there. That seems to be very popular.

So I just wanted to drop those sentiments on here, that we see elliptic curve really coming on in Europe more so than anywhere else. All of

them are dropping RSA SHA-1 off the table. And that's where we are with the choice of algorithms.

And I will end there and take questions and then go to Viktor.

EBERHARD LISSE: Thank you very much. It would be interesting to see how this goes by signing platform because I think a number of African TLDs, for example, are signed on the same [inaudible]. So if one changes, the others change.

Anyways, thank you very much. We're quite sure that you will do this again because over time it will be interesting to see how this changes in a year or so.

EDWARD LEWIS: Yep.

EBERHARD LISSE: Okay, next will be Viktor. You have he floor. We can hear you and see your presentation.

VIKTOR DUKHOVNI: Okay. So I guess you're driving the slides. That's fine. Okay, so I want to talk about two aspects of DNSSEC parameter choices. One is of course that we want to operate DNSSEC in a way that's practically reasonably secure so that we don't present weak DNSSEC parameters to the world.

But I think we want to go a little bit more than just good enough in practice.

Because I think DNSSEC, to be really useful, needs to not only be reasonably secure but to be trusted in that people are increasingly publishing information in DNSSEC that we want to be able to rely on like the new HTTPS records that provide security-relevant data when people are accessing websites.

And there's also some privacy data that published in DNS nowadays and so on. And CAA records that protect the records that protect the issuance of certificates. All of these should be signed, but also people need to have confidence that they're reasonably tamper resistant. Okay, next slide.

So I terms of the implementation status of DNSSEC algorithms and which ones are required and which ones are recommended and so on, we have RFC 8624 as a good reference. We may yet update it in the coming years, but that's the currently state of the world.

And some of the algorithms, especially the SHA-1 ones, as mentioned, have now been deprecated, so algorithms 5 and 7 should be rolled away from existence as soon as reasonably possible. And there's a nice website up that Tony Finch [wrote up] that explains why SHA-1 is no longer good for DNS.

The best practice algorithms and the ones that are mandatory to implement in RFC 8624 are now eight, which is RSA with SHA-256 and

also 13 with ECDSA, the prime 256-bit curve. And SHA-256 is a signature. It's algorithm 13.

Now I keep talking about doing rollovers and improving DNSSEC and so on. And the first slide I'm going to show you is that, in fact—maybe it's the pandemic or not—lots of progress is being made. DNSSEC isn't stuck in the 2010 era.

And in particular—next slide, please—I do a lot of surveys for not just TLDs but in fact the domains immediately below the commercial delegation points, as they were referred to earlier. And so this graph shows you what algorithms are in use, not by TLDs but by the delegated subdomains—the 16.4 million of them at the moment in my survey.

And what you see is that the green and the red algorithms, 8 and 13, which are the recommended ones are growing by leaps and bounds. Algorithm 7 which is the next most popular algorithm has recently taken quite a dramatic drop. That's the light blue, and it's almost disappeared. And most of that growth has gone into contributing to the growth of algorithm 8. People stayed with RSA but migrated from 7 to 8. Some of them migrated from 7 to 13, so we see both of them growing quite nicely.

Algorithm 5 largely disappeared already more than a year ago. There are about 30,000 of these domains left for algorithm 5, and those last 30,000 are pretty stable if you can see the flat line just about the X axis there.

And we have a tiny population of algorithms 10 and 14 which are actually quite strong, but just sort of more strong than you need and not very popular.

But in any case, migrations are happening and so rollovers are possible and taking place in the e-TLD+1s, the end user domains if you like. Next slide.

So here's the story again with TLDs. Not in graphical form, but just a table. I'm showing that among TLDs, algorithms 5 and 7, there are just a few TLDs left—29 and 39, respectively. By far the majority of TLDs are on algorithm 8. That's the gray on Ed's slides. A few, 44 of them are on ECDSA-P256. I'd like to see more of that in the coming months and years. And then we have a few that are even on algorithm 10 which is just fine. I'm recommending 13 and 8 if you're stuck with RSA. Next slide.

Okay. Because we want DNSSEC to not only be modestly secure but actually trustworthy, I think it is now time to move beyond the 1024-bit RSA which is largely used as the zone signing key algorithm by most TLDs. A batch of TLDs have moved to RSA-1280 bit, and we'll talk about that more soon.

But 1024-bit RSA is broadly criticized by the web people as kind of … That's why they don't like DNSSEC because it's so weak. It's still using 1024-bit RSA which NIST recommended to be phased out by 2010. We're now 11 years on, and we're still using 1024-bit RSA widely in DNSSEC.

So in terms of what are the risks, already in the last year, in February [2020]—so more than a year ago now—I don't know if you were paying attention. There were various other more interesting events going on around that time, like the start of the pandemic. But there was progress in RSA factoring where an 829-bit key was factored and took 2700 core-years. And there's a link in the slide, too, if you're interested in the details.

I will extrapolate from there to estimate the costs of factoring RSA 1024-bit and various other strings of RSA keys. Just by scaling the NIST formula for the costs of factoring large numbers from the time it took to do this challenge to the various keys of interest. Next slide.

So first I'm going to look at the key-signing keys. This is a story that for most TLDs is in pretty good shape. TLDs generally have strong KSKs. We see only two TLDs that have rsa1024 for their key-signing keys. The factoring cost is about 2 to the 80, and it can be done in half a million core-years which, though a very large number, in fact is not outside of the resources of a nation state or Amazon or a Google cloud provider and so on. Could in principle, if they wanted to, expend that much computer power at some cost and crack our, say, 1024-bit keys today if it were of interest to them.

By far, the majority of the TLDs are at 2048-bit keys. 1,300 such KSKs. And no classical algorithm can come anywhere close to cracking those. I'm not even bothering to extrapolate the number of core-years. We need radically different algorithms to attack 2048-bit keys.

Some TLDs use 4,096-bit RSA keys. I rather think that's overkill. Unnecessarily strong and unnecessarily large key sizes. Almost nobody benefits from this. The operating systems that we run on verify software updates and such with weaker keys than this. You're unlikely to be benefiting from overkill of this sort.

And 45 TLDs have adopts ECDA, and I'm quite pleased with those. Those also are very, very strong and would require brand-new factoring technology to attack. Next slide.

Zone signing keys. This is where I think we have, by far, the most work to do in the DNS space. If you're sticking with RSA, especially with NSEC3, I'm recommending 1280-bit keys due to packet size limitations. RSA-1024 is used by 804. So by far the largest number of TLDs are using 1024-bit zone signing keys.

As I mention, they can be factored in, at most really, half a million core-years. If somebody had better algorithms that are not published or can use parallel approaches that perhaps Dan Bernstein and others are suggesting, they may be able to lower the cost on a per key or an individual key basis.

The algorithm has reasonably fast verify and signing performance, but already it's NSEC3 packet sizes are around a kilobyte pushing towards the recommended UDP sizes of 1200 bytes. We're getting close.

RSA-1280, which is now used by .com, .net, .org, and 618 TLDs at this point, has a narrow but I think quite significant factoring cost margin. It's 89 bits or 240 million core-years. I don't see … If that's a realistic

estimate, if there aren't much faster algorithms that we know in public, then I don't see anybody expending 240 million core-years to crack at RSA 1280-bit key. It is practically, reasonably secure.

It's NSEC size is 1207 bytes, typically medium packet size from querying the actual TLDs which fits within the recommendations for UDP larger sizes. So it's not an accident that .com and .net and so on chose this particular RSA key size. That's kind of the largest you can reasonably use without causing TCP failover for NSEC3.

The next key of interest is RSA-1536. It is by far not popular with TLDs, and none are using it. However, it is quite strong. 54 billion core-years to factor and still has reasonable performance. It's NSEC3 size now goes over UDP limit, so you wouldn't use it with NSEC3.

However, it's NSEC size—which I didn't measure from any actual TLDs because none have deployed it—is less than it would be with RSA-2048. So it's under a kilobyte. So it would be quite practical for NSEC3 to use RSA-1536 if you wanted to choose a compromise key and wanted it slightly faster at signing and verification.

However, quite a number of TLDs, 162, are using RSA-2048. Definitely strong enough as before for the KSKs. And as we can see, the NSEC3 size is already definitely too big for UDP, so we're looking at TCP failover for NSEC3 for negative responses. For NSEC, however, we're still under 1200. So either 1536 or 2048 is a good choice for your ZSK if you're using NSEC.

And 2048 would get key size parity with the WebPKI. So if you wanted people to trust your DNSSEC as much as they trust the WebPKI, that may be a choice. Although in practice, 1536 is equally good.

And then finally, of course ECDSA-P256 is, I think, where we are until quantum computers arrive some two or three decades from now. This is the best practice, though not yet widely adopted. Next slide.

Here I'm showing the DNSKEY response sizes. This matters, oh, not so much for negative responses but when resolvers are first loading [the] DNSKEYs before they can verify any data in their zone. If we want to also avoid TCP failover there, we see that ECDSA, the P256 DNSKEY responses, are 500 bytes or less.

But with RSA we see a few bumps depending on which sizes of RSA keys you choose. They're either close to 1K or close to 1500. Or there's even a long tail heading towards 2,000-byte response sizes. So I think this again shows the advantages of using ECDSA. If you're going to rollover to something instead of changing RSA key sizes, it may be more sensible to choose ECDSA. Next slide.

So I'd like to strongly encourage ccTLD and other TLD operators to move away from 1024-bit ZSKs to at least 1280 if you're using NSEC3 or 1536 potentially if using NSEC—maybe even 2048 if you're using NSEC. Both stay under the UDP packet sizes that you can realistically achieve if you're using NSEC.

If you're using algorithms 5 or 7, there are very few ccTLDs in that boat. I found just four. Then definitely move to algorithm 8 If you're stuck with

RSA. You can use algorithm 10. It's perfectly fine, although not popular. Pick one or the other.

Ensure that your KSKs is 2048-bit. Do not use 4096. It's needlessly large. It just bloats packet sizes and adds no realistic security.

If you are using 1280 or even still 1024-bit keys for a while until you migrate, do rotate your ZSKs regularly. These are not necessarily out of reach of being brute forced, especially if our estimates for the cost of breaking RSA are too conservative and somebody knows a faster way of doing that than has been published.

In terms of rotation practices, I'm seeing 135 TLDs that are still using 1024-bit ZSKs that have been around for significantly more than 90 days. So now for almost nine months plus. So 16 of those are TLDs, so significantly .uk. And Estonia and Vietnam and China and Greece are all using rather stale ZSKs. These should be rotated, ideally.

And on the other hand, 6,638 TLDs that are still using 1024-bit RSA have recently rolled their keys, at most four months ago. Although four months is already over 90 days, so some of those will be nice to see rotate as well. Next slide.

So if you do have the luxury of moving away from RSA, definitely use algorithm 13. It is by far the most effective in terms of packet sizes, signing performance of signing the zone, and so on. The only downside is that verification is a little bit slower with algorithm 13 than it is for all the others. But they already account for about half of all of the e-TLD+1

zones, and the resolvers are handling ECDSA just fine. So there are no known disadvantages.

It's all positive except for availability of software stacks and tools which are there. It's just that you may need to upgrade your software and your hardware to take advantage of this algorithm. It is mandatory to implement. It's got smaller DNSKEYs, faster zone signing. And keys are definitely at last as strong as WebPKI.

If you're not too worried about zone walks, especially if your zone data is already public and so on, consider NSEC instead of NSEC3. There are lots of advantages to NSEC.

And if you're signing on the fly, sort of like Cloudflare does with many of their customer domains, then NSEC doesn't even leak any data because you publish narrow zone signing denials that don't enable zone walking. That's really the most secure way to not leak your zone data. No hashes to crack, unlike NSEC3.

And of course, with NSEC3 which is a different draft that Wes and I are working on, if you are using it—and TLDs have largely already addressed this, so I'm just repeating it—use low iteration counts. All TLDs are now 25 or less, so really, thank for everybody who took care of that. The remaining high iteration counts are largely only in the e-TLD+1 space.

So with this, I'd like to encourage everybody to, over the next year or so, improve DNSSEC algorithm hygiene. Send questions to DNSOP or DNS

Operations lists and we can help you migrate to a more trustworthy and non-deprecated DNS. I'm ready for questions.

EBERHARD LISSE: Thank you very much. I can take one or two questions because we're already into the break time and staff needs their break as much as some of us do. Any hands?

There were some remarks in the Q&A which were not really questions or are already answered, so I did away with them. If somebody wants to ask the question, please, now is the chance. I'm not seeing any hands.

So thank you very much. This was a very interesting presentation as well, and quite helpful I think, and fit nicely with what Ed did with the graph. In the future, you must maybe combine this and do this on a regular basis once a year so we see where there are changes.

Anyway, thank you very much. There's something in the chat. Hang on. Thank you very much, and we'll see each other in half an hour again.

BRENDA BREWER: Thank you. Please stop the recording.

[END OF TRANSCRIPTION]