ICANN70 | Virtual Community Forum – Tech Day (3 of 4)
Monday, March 22, 2021 – 12:30 to 14:00 EST

KIMBERLY CARLSON: Hi, everyone. Welcome back to Tech Day part three. For the sake of time, any reminders and housekeeping items will be placed in chat. Please note, with this session being recorded and follows the ICANN Expected Standards Of Behavior. And with that, I'll turn the call back over to you Eberhard. Thanks.

EBERHARD LISSE: Thank you very much. We are not strapped for time so we can do these things and give everybody time to have a look at the chat. In any case, can you bring up the block three and four slides? The previous slide, please. Thank you. We now are going to hear a little bit from Brian King and Brian Lonergan. Brian King is from MarkMonitor. Brian Lonergan is from Donuts and they will speak about homoglyph domain names.

BRIAN KING: Thank you very much. And it's my pleasure to be joined here with Brian Lonergan today. I would like to first apologize. I was supposed to give this talk during the last ICANN meeting and due to a scheduling confusion on my end, mixed this call up with the gTLD Tech Ops call and wasn't available. So, I apologize for that but I am excited and hope that this presentation benefits from a few more months of preparation and a little more research. So, thank you for having me here today.

I'd like to have a conversation and so I would encourage folks to raise your hand and to ask questions in the chat as we get into some of some of what Brian Lonergan and I have been discussing with regards to homoglyph or homograph domain names and the vocabulary changes. A lot of people use the terms homoglyph and homograph interchangeably and we'll talk about what we mean in a minute. Essentially, these are domain names that are confusingly similar in appearance although not identical in the DNS. So, Kim, if you're in control of the slides, could we go to the next one, please? And then one more, please.

Thank you. So, what we have on the screen here today are all different domain names and several of them don't look so different from the others. There are no spaces in these domain names. Some of the characters just represent visually with a bit of space. And you'll see on the top right-hand corner there a domain name that consists of entirely of those characters that look like Latin characters represented in ASCII, A to Z type characters. However, these are all IDN domain names and we'll talk in a minute about—actually we should start now on talking about why these can be problematic.

So, in the domain name system, registries and registrars like to do what they can to address technical forms of DNS abuse—things like phishing and malware. And what we've seen or what I've recently come to learn about these types of domain names is that although they are IDNs and are actually representative in Punycode by the XN-- prefix, these display natively in email clients and browsers in some cases and can be really

confusing and we think especially impactful when they're used in phishing attacks.

So, some of these are prohibited by registry policy from being registered, which is a good thing but some of them are not. And these particular domain names just call out .com but really the concept here applies to any TLD. And we'll talk a bit about some registry policy decisions that are available to registries. Brian, do you have anything to add about these particular domain names before we show which ones are available?

BRIAN LONERGAN: No, I think you framed it correctly.

BRIAN KING: Excellent. So, Kim, if we can go to the next slide, please. So, current .com policy does a good job of preventing the registration of all the domain names that are not bolded. So, the domains in bold character are homoglyph variants that are available to be registered in the .com space. I'm not picking on Verisign here for the .com policy. In fact, we applaud Verisign in a couple more slides for how they've approached this problem.

But I was surprised to see that the three of these variations, and really just using the different variations of the letter A there, make these three distinct domain names that, at least according to my registrar system, will let me submit a registration command to Verisign for these. So, are there any questions about that so far—the concept, why it's potentially

problematic? Okay. I have an eye on the chat and the hand raise feature in Zoom if anyone does. I have a question in the chat. I wonder if Brian—

EBERHARD LISSE: We'll take the questions from the question and A pod so you don't have to be interrupted all the time for your thread. If you want to interrupt, when you see a hand, finish your thread and then we can go there. But the point is everybody having questions should put them in the Q and A pod and then we do it at your convenience [inaudible].

BRIAN KING: Very good. Thank you. I think that makes sense. I can address that the question in the chat, if I implicitly refer to Cyrillic based IDNs. No, not exactly. We'll get into some different policy options in a moment. But what I've seen primarily as a risk here are characters within the Latin character set that are confusable with other Latin characters, or the very basic Latin characters that we use in the English language. So, we'll talk a bit about policy and why the concept of Cyrillic characters is less of a risk now.

Let's actually move on to the next slide so we can get into that. Okay. So, these are other variations and just, I'd like folks to take a look and see which of these they think are homoglyph variants and which ones are the standard ASCII characters. If we were in a room, I'd ask folks to raise their hand for which ones they were. But folks can generally see in the bottom right that the two O's have moved closer together, right?

And then the bottom left as well. To my eye on the top right, the L looks a little off.

You have the advantage today of seeing all of these close together so you can kind of compare side by side. But imagine in a phishing attack or an email that you receive where this is a standalone domain name and perhaps those two O's that are slightly closer together. Maybe that doesn't look so peculiar, as I see in the chat, if you see that as a standalone domain name. So, Brian, correct me if I'm wrong but I think these are all actually homoglyph variants and none of them is actually the google.domains domain name. I think we pulled all of these from the Donuts spinner.

BRIAN LONERGAN: That is correct. And they're actually all coming from the Latin script table that both Donuts and Verisign use today or very similar versions of that Latin script table. And both script tables contain in excess of 600 characters, right? So it's not specific to Hebrew or Cyrillic or any other character sets. You can have problems inside a single script table.

The one other piece I'll call attention to is that in some cases where it may appear pretty noticeable here—those two O's close together or perhaps the L in the top right example here—depending on what service is rendering and the domain name, whether that's an email client or a browser and fonts, etc., these can become more or less pronounced in terms of how confusable they are with the anchor domain name.

BRIAN KING: Yeah, that's a great point. So, if we go to the next slide, please. These are the Punycode variants. So, this tells you the actual domain name so you can see how easy it is to be fooled by these. Looking at the last slide, I thought maybe, two or perhaps three of them were the actual domain name.

So, if we go to the next slide, Brian captured the complicating factors, we're not just—we here in the ICANN community focus on domain names and their place in the domain name system. But out in the real world, domain names show up in browsers, in emails and in places where different fonts can be used, different sizes and colors of letters can be used to perhaps hide some of the dissimilarities where they may appear in the domain name space.

If we go one more slide and then again, one more after that, I'd like to talk a bit about where current policy is now in the gTLD space, understanding that ccTLDs are their own world when it comes to policy. But I thought this group might benefit from an overview of where gTLD policy is on this and then the types of policy decisions that other registries have made with regard to homoglyph variants and why those might be appropriate for both gTLD and ccTLD registries.

So, here, ICANN has guidelines for IDNs. Version three is the current controlling version. It was published in 2011, so 10 years ago. There's been quite a bit of research, quite a few phishing attacks and a lot more thought put into IDNs since then. It's important for me to say that, MarkMonitor and I think many or most or everyone here is—we are proponents of IDNs and we want to encourage the domain name system

and the internet community to use IDNs, to enable internet users to communicate and use the DNS in their own language, whether that's a Latin-based language or not.

Within the ICANN guidelines, Unicode was the agreed standard for domain names. That's a native-to-technology type thing and the Unicode has had a lot of policy developed independently of the ICANN community for years. So, it was agreed to be the standard.

Within the ICANN space, a domain name can only contain one script. So, you can't have a domain name that's made up of characters from a Latin language and Arabic, for example. You can't have a domain name that includes both of those, or Arabic and Chinese, or Cyrillic and Latin. So that's the kind of baseline.

What remains though are the concept of whole script confusables. That's a great clarification from Michael in the chat. It is one script per label—correct—to the left side of the dot. And that those whole script confusables remain available for exploitation. So, these are confusable characters that exist within the same script and a script may contain multiple languages. So, I've been talking for a while. Let me ask Brian to maybe elaborate a bit on the difference between a script and a language and what it means that those whole script confusables remain available.

BRIAN LONERGAN: Sure. And this is kind of something that wasn't massively controlled by policy, either in the original round of new top-level domains or

especially again with new gTLDs which is, when implementing IDNs, registries can choose between covering multiple languages in a larger script table—usually Latin and using it to manage languages like English, and French, and Spanish, and German that use a common Latin script table. Or your registry backend can define language tables specifically and have individual tables for French, and Spanish, and German. And that's true of Indian languages and Devanagari and it's true of Arabic script and going more defined in that level as well.

So, where you have a larger script table, such as the Verisign and Donuts Latin script table that has hundreds of code points, as opposed to 22, or 26, or 30 in some languages' sakes, it provides a little bit more lateral movement to create confusable looking labels using the same authoritative language script. And that provides a bit of an opening. That is where the majority of the issues arise today is large language scripts being used in a non-mixed script registration to create confusingly similar domain names.

BRIAN KING: Yeah. Thank you, Brian. If we can go to the next slide. We talk about how various registries … And this is primarily a registry matter, setting policy for the entire top-level domain which is why I keep saying "registries" here. There are a few different things that the registries can do with their policies to help address this problem and hopefully prevent the registration and use of these kinds of domain names in phishing attacks.

The first is the baseline required by ICANN—that we talked about—required by ICANN for gTLDs and that is the prohibition on mixed script

domains. So, that's a good policy. It's good that we have that and that prevents the use of a Cyrillic A, for example, in an otherwise Latin character domain name—or label, I should say, as Michael correctly clarified in the chat.

Some other options that are available above and beyond that … Really, we do see that as a baseline. Above and beyond that, registries could and many registries do—some registries do—require that when someone submits a domain registration that within that EPP command, they include what language the domain name was in. And what this does is it eliminates, as Brian mentioned, the possibility of a domain name registration that contains characters from multiple languages within the same script. So, it prevents the registration of a domain name that has a character that exists only in Spanish, for example, with a character that exists only in French. And it prevents domain name registrations that contain those variations within the label.

So, today, without that, one could register a domain name that contains both Spanish-exclusive and French-exclusive characters and just tell the registry the script is Latin, both of which Spanish and French are within the Latin character set, and get that registration through today without this, if the registry didn't have this additional policy choice.

BRIAN LONERGAN: I'll just offer one alternative there, Brian, for why that exists today or what you may choose a language script as opposed to individual tables. And that's kind of historic registrar and registrant behavior. The downside of moving to individual language tables for every supported

IDN from a registry, is that now you're requiring both the registrant and the registrar to explicitly nominate the table that they want to use for a given registration, right?

So, if I want to register a domain in Spanish, as a registrant, on my registrar's page, I need to go through a dropdown and select the character codes and do that correctly. And that registrar, then, must pass the character code at a registry in order to have a positive registration. Larger scripts like Latin, Devanagari, Arabic make it a little bit easier because they removed that requirement from registrants and registrars and give a little bit more lateral movement in terms of ease of registration for variety and domain names.

BRIAN KING: That's right. Thanks, Brian. And what that does by not requiring that language tag is that in Latin, for example, the number of characters within that character set has grown since 2011. And now it does include more confusable characters. And without requiring that language tag, then those exist within the same script.

So, a bad policy choice and one, to be clear, that I'm not advocating and I don't think Brian's advocating either, is to block those similar—they're called confusables—to block those confusable characters outright. This is a policy choice that registries could make on their own. But what that does is that, unfortunately, limits the use of those characters in any domain name. So, this works across purposes with the concept of IDNs, which should allow non-English-speaking registrants the ability to

interact with the DNS in their own language. So, that is a policy choice that we want to be clear is not one that that we're advocating for.

But one that Donuts does, and which I really admire and encourage, is this last one of blocking variants and confusables after a domain name is registered in ASCII characters. And Brian, do you want to talk a bit about how Donuts does that?

BRIAN LONERGAN: Sure. And the decision process came from some thoughtful discussion internally in our business. We did not want to remove, again, registrations from our business model, right? It's a very valuable piece and we also believe in localizing the internet where people can make registrations in their own vernacular. We also didn't specifically want to move to individual language tables overnight and away from our larger script model, primarily because of the overhead it would create for registrars and registrants and the confusion that would arise. But we did want to protect our user base, new and existing, from this type of homoglyph phishing.

So today, we've developed an algorithm which will, at the point of registration spin all known—I guess, all known within the Unicode confusable table—permutations of homoglyphs for your domain registration.

So, if you take the example of, I want to register brian.news, we will then take Brian and spin the B, and the R, and the I, and the A, and the N against the Unicode confusable table and create a couple of hundred

permutations of that domain name and remove them from the availability pool from our registry. So, when you purchase an ASCII registration, you now have the authentic ASCII registration and we'll prevent registrations that look similar to it retroactively. And all of that happens in real time.

It's something we've been doing for, I think, probably about 12 months or so now and we've had success, primarily from registrars in the corporate space who've seen the benefits for their clients—clients who historically had been spending a lot of time and energy making defensive registrations against our TLDs and other TLDs in the space because they understood the vulnerability existed. And we still continue to see fail check and fail create registrations hitting our core SRS today, roughly at a rate of about 10,000 a month queries do we see for checks or fail creates against blocked domain names—domains that have been blocked due to our homograph policy.

BRIAN KING: That's great, Brian. Thank you. I see a question for you, I think in the Q and A. And it may be just clarifying how I wrote up the slide. Is it only after an ASCII registration or is it after any registration that Donuts spins up those confusables and then blocks them?

BRIAN LONERGAN: Yeah. That's a great question. So, so we do not do it in reverse today. Today, it's only based o … The anchor registration, I guess—the first registration—must be based in ASCII and not an initial IDN registration.

So, we will not take an IDN registration and block the resulting ASCII label in reverse of that. The majority rule for that is essentially because most of the internet is written in ASCII, right? IDNS are pretty new concept and they make up a pretty small percentage of our backend registry services and also those of .com, right? It's a fraction of the community. So, we haven't decided to do it in reverse.

And we also, at the time of testing, did not have any example of someone creating permutations of IDNs in ASCII for malicious causes. So, we weren't able to track it as a vulnerability. Now, should that come up in the future, or we do begin to see behaviors that look like that, it's something that we might reassess.

BRIAN KING: Okay. Thank you, Brian. Great question. So, to the other options here, I promised that we would we would be clarifying—or that we would be congratulating Verisign and encouraging their approach here so this is where we'll do that. There's another option to find a middle ground here. The registries can stay with that baseline approach and just follow the ICANN policy but then also proactively identify confusable characters and then block those after an ASCII registration. So, this is what Verisign has done. And the link is in the in the presentation here, to the article talking about what Verisign did. They identified a few confusable characters that they now block when the ASCII domain name is registered. I think there's probably more.

BRIAN LONERGAN: Sorry, Brian. Let me just cut you off. I'll just make a minor correction. I actually believe what they've done is remove three characters from their approved language script which they have decided are particularly vulnerable to this type of behavior. Again, it's a smart move but with the downside of it does limit the character set available for registration.

BRIAN KING: Got it. Yeah. Thank you for that clarification. That's useful. But the nuance here is really important as I think has become obvious to a lot of folks on the call. Okay. And Jothan has the link now. Great.

Another option that we've seen that goes well beyond even IDNs and homoglyphs is what UNR, formerly known as Unit Registry, has done. And they do it with their EPS block, I believe. I don't think they do it as the domain registration or availability policy but they also block and spin up what we call the leet-speak approach. And this is using numbers and other characters that look like Latin domains and they block those as well. These are the ways that hackers historically have typed to avoid natural language spying from folks who might check out what they're up to. So, they go above and beyond. That's right, Frank, in the in the chat. So, that's another way that registries have developed policy and made a choice.

So, I think we're ready to go to the next slide and then the next one after that. So, how big is the problem? And the answer, I think, is unknown at this time. We do know that IDN domain names and homoglyph, homograph domain names are a low percentage of current phishing domains but that's just a percentage. So, we have seen these used in the

wild. There's a couple of examples here that were actual phishing attacks that we've included in the slide. So, it's a low percentage today but, I think if one phishing attack could be prevented, then it's worth taking a look and considering a stronger homoglyph, homograph policy.

And the thing that I'd be interested to read more research about, if anyone had done it or help do that research, is the impact of these phishing attacks. And we believe that these homoglyph, homograph domain names are more believable and therefore potentially far more powerful phishing domain names that might have a bigger impact. But I think studies would need to be done to determine whether the increased believability of these versus an uglier phishing URL do actually perform better—not better—perform more effectively in phishing campaigns. I think we'd welcome more research on that. Is there anything that you would add here, Brian?

BRIAN LONERGAN: No. I think that's exactly right. Yeah.

BRIAN KING: Very good. Okay. So, I do see another question—two questions now in the Q and A. And let me take a look at those.

EBERHARD LISSE: Carry on with your presentation. I don't really find the interruptions very helpful. We can sort this out at the end. We have time so don't worry.

BRIAN KING: Okay. Thank you. Well, we are at the end. If we go to the next slide, we have thanks to Donuts, to UNR and Verisign who have provided me with insights and guidance on this this problem and provided a forum for discussing potential policy solutions that can address this problem. I would like for this to be the beginning of a conversation that I hope will be ongoing about how we enable IDN policy that works well for the non-English speaking users of the DNS but which can also prevent the types of harm that we're seeing here that can come from abuse of these policy outcomes. So, we can get to questions now. If we go one more slide, I think is the questions slide. It's the same blimp, same weather balloon.

EBERHARD LISSE: Okey-doke. Thank you very much. Unfortunately, Maciej Korczynski from the COMAR Project that presented earlier, asked to present earlier because he couldn't be available late in the evening. And I'll bring you guys into touch because that's … I would like to see what he says about his detection of malicious domain names and malicious websites—whether you guys with these IDN homonyms and homographs can work on that. Okay. Let's take the first question and we have the Q and A part open, Brian, so you can read it yourself. Read it out loud and then answer it.

BRIAN KING: Sure. I'd be happy to do that. So, RFC8753 points out the review process within the IETF when a new Unicode version is released. It very well might result in code points that can be added to explicit exclusion instead of calculation that would result in P valid as derived property

value. Is what you do implicitly or explicitly send a message to IETF to be more conservative in the review of new versions of Unicode? That's an interesting question. My reaction is to kick that to Brian, who's a lot smarter than I am on this stuff. Do you have an immediate reaction, Brian?

BRIAN LONERGAN: It's a great question. My gut feeling reaction would be probably not. We are not defining new confusable characters. In fact, the Donuts service that we use for spinning permutations actually relies on some work that Unicode did in 2015 called the IDN confusable tables. And that contains a couple of thousand characters which Unicode defined to be confusable with each other. So, although we've essentially productized that work that they've done, it's unlikely that it's having an impact or causing them to be more conservative than RFC specifications.

Typically, I expect that to be used, not from a not from a malicious point of view but normally from like a collision or confusable point of view. So, one example would be a double S or sharp S in German, where the character is actually has the same meaning in both sides and doesn't just look confusingly similar but is actually used interchangeably. And those are rules that most registry backends will implement as a standard behavior outside of this more objective blocking behavior.

BRIAN KING: Yeah. Thank you, Brian. I would also say that I think we might be wise to take advantage of the good work that the professional linguists have

done in developing those confusables and in working with Unicode. So, the next question I have is, how do you provide language/language tag information to the end users through WHOIS/RDAP? I think that we need mechanisms to improve human perception.

That's a great question. I know that EPP can facilitate that language tag between the registrar and the registry. And when registries require it, registrars would have to send it through. But if your question is about how that's passed to the end user, I don't have an answer. That's a really interesting one that my gut responses may be that the browser forums might have a piece of the answer there, if they know—the browsers know which domain names are supposed to be in which languages or the email provider's consortium. Brian, do you have any immediate reaction?

BRIAN LONERGAN: No. Again, that's a good question. I think in most circumstances, WHOIS will respond if you create either the Unicode or the Punycode version, right? So, simply searching for the XN-- version of the same domain name would allow you to identify it.

I will say that from a browser perspective, both Chrome and Firefox have kind of flip-flopped on this behavior a couple of times. In some scenarios in the past, they used to translate any IDN registrations into its XN—format—into its Punycode format—and render it that way so you realized you were hitting a non-ASCII domain registration. And they've kind of gone back and forth on what's best practice there. And there's also a bunch of extensions in the browser world and the email client

world that will allow you to expose whether the domain you are landing on is in ASCII or is a IDN domain registration.

BRIAN KING: Yeah, that's a great response, Brian. And thank you so much for the question. That, to me, feels like one that where there's great potential for collaboration and work to be done. So, thank you very much. I have another question, “Were the examples you showed using a fixed width font? And what impact do font choice and kerning have on label disambiguation?”

Font definitely has an impact on how easy it is for humans to perceive a difference or that there might be something fishy—pun intended—about the particular domain name. I don't know if it was a fixed width font in the PowerPoint but I'm aware that that can be helpful or harmful. I think what you're talking about are fonts that were developed, or at least certainly benefit people with dyslexia, that show that show characters with a fixed width so that an I doesn't need to be skinnier than an O but that each character has its own width. It’d be an interesting thing to study for sure. Any thoughts, Brian?

BRIAN LONERGAN: No, I would agree. I guess the base answer is, yeah, it does have a pretty big and significant impact on how confusable a character set may appear. Typically, it doesn't have such a huge impact in your URL bar but when rendered in regular font via your CMS or wherever you're

hosting your content, it can have a really big impact on whether a domain appears confusable with its anchor or not.

EBERHARD LISSE: If I may get in here quickly, as I said before, most of you know what I do for a living. I'm a gynecologist. And we have to do a regular meeting every week where we have to do journal presentations. And I did once one, what you can do wrong with PowerPoint. And one of them is fonts. And we then looked at 10 different fonts—italic, non-italic script, Roman and fixed width and so on—to decide one which is best on screen for our … We are all over the age of 50. Most of us are around 60. And we came up with a font that all of us agreed we can read, which is actually different from the one they wanted to have the paper presentation. Yeah.

So, it's actually quite important this question that when you get a funny looking link, don't click it but how to address the situation programmatically which is what you have to do. If you have largescale registrations, it's really a very, very interesting question. Carry on with the questions. We have time.

BRIAN KING: Thank you very much. That's a great insight. And I'm seeing some good comments in the chat as well—I'd like to follow up on probably offline—regarding the Latin GP and the IETF work on Unicode 12 and 13 specifically. So, I'll get back to those folks on that.

The next question in the Q and A is, “Are you in contact with the universal acceptance steering group to discuss the problem and craft solutions? So, I meet with the UASG every so often. I would typically try to attend in-person during ICANN meetings and would love to, again, sometime soon. I haven't had any formal conversations with them about homoglyph variants but I'm familiar with their very important work and I am supportive of UASG. Believe Donuts is as well. I know that the folks there follow that work. So, that's a good question. Anything to add Brian about UASG?

BRIAN LONERGAN: No. It's a great call out. Yeah, we are members as well and participate on different conversations but actually haven't explicitly had this conversation with the UASG Group. So, it's a good point. Something we'll look at it.

EBERHARD LISSE: All right. I don't see any more questions or raised hands. Thank you very much for this presentation. Was a bit of tricky to round you guys up but that's [inaudible] for you, in the end. I'll punish you for missing last time but it was a great presentation and it was really worthwhile chasing this up. So, thank you very much and feel free, if you've got the similar stuff, to get in touch with us.

BRIAN KING: Very good. Thank you. It was my pleasure.

EBERHARD LISSE: Unfortunately, we seem to are missing the host and the host's presentation so we are going to move straight on. If you could put the … Yes, that's the one. We're going to move straight on to the next topic which actually fits quite nicely in what we heard just now. And Champika, you have the floor.

CHAMPIKA WIJAYATUNGA: Thank you. Yeah. Hello everyone. I represent ICANN Org. There are a lot of good discussions happened in the previous presentation, especially the last one. So, my presentation is more of a brief tutorial that we wanted to give, especially for people who want to get more understanding—awareness about the email address internationalization. So, as I said, there are some overlaps when comparing with the previous presentation. So it's good in a way so I can skip some of those things quite quickly as well. So, let's move on. Next slide please.

Okay. Next one. Yeah. So, before we really get into email address internationalization—or simply, we call EAI—you can just see the typical ASCII domain name, what we have. I think there were a lot of discussions, as I said earlier. So, there are some certain rules involved here. For example, when you think of the top-level domains, when you think of the ASCII label, we can only have letters over there from A to Z and the label length is 63. And then, when you consider the second levels and the third levels and so on, we can have the letters, digits and

hyphens. And so those are constraints, the LDH rules, basically. Next slide please.

Now, we have been, of course … When you consider the ASCII with regard to that representation that we were discussing, when you try to represent the domain names in ASCII levels, as we know, obviously now the ASCII representation, the ASCII table has just a snapshot of that. And we only have 137 values that we can use but we don't use all of that. The highlighted ones that you can see in green, that those are the letters and then the blue outlined ones are the digits, and then we have the hyphen. So, these are the ones that we use in ASCII. Next slide please. And then for top-level domains, as we discussed, we use only the letters. We don't use digits and hyphen for the top-level domains. Next slide, please.

Okay. Now, when it gets to the internationalized domain names—so IDN labels—the rule is that obviously, we have to … Now, we are not using the ASCII labels here, the A labels. We are using the valid U labels. So, we still have that letter principle for the top-level domains. And then also, actually for the second levels and so on, we have the Unicode code points that is actually constrained by the LDH scheme that we discussed. And this is within the IDNA 2008. So, that's the standard we use, the IDNA 2008 standard. Next slide, please.

Okay. Now, we also actually have to understand the Unicode encoding here because earlier I mentioned to you that the ASCII, of course, we have a limitation and we can represent the Latin base, the English character set but then if you want to go beyond that, the local language

scripts and so on, there isn't space over there so we have to consider other encoding schemes if you want to do that.

So, now, what we do use here is actually the Unicode and so here we encode the glyphs into code points. And so, when you try to cover various different scripts, we use different code points. So, what you see in this slide, just a snapshot of the Unicode table of the Arabic scripts. And then these code points are shown in [Hexa] using this notation. But you see with U+ and that's the code point that we show.

And now these code points, they are typically carried using the UTF-8. So, UTF is basically the Unicode transformation format which is eight bit. Now, again, with UTF, there are different other schemes as well like UTF16, 32, and so on. But UTF-8 the most optimal because for different reasons, like you can have variable number of bytes in terms of representing the glyphs. For example, it can be one byte, two bytes, three and so on. And also, actually, it is backward compatible with ASCII. This is also something very important because we don't have to then do any changes with what we have been doing with ASCII. So because of that, it's the main sta,ndard for carrying the Unicode code point in a lot of these web protocols and so on that we use now. Next slide.

Okay. Now, we also have to actually do something called Unicode Normalization. This is because when we try to represent, say, a certain glyph, now that glyph can be represented in different ways. So that can be given in different ways. Say, in this example, you can see an E with an accent, which has the code point 00E8. And also, that E with the accent can also actually be given with a sequence, where you have the E and

the accent in separate code points as well, with this 0065 and 02CB, actually. So, the thing is that we can't use two different representations here because it should be the same glyph that we are talking about.

So, because of this, we have to do something called normalization. And then there are different normalization schemes. And what we use here is something called normalization form C. That's the type that we use. And also, actually, the case folding is also not very stable here because say, for example, in this case, the small e with the accent, this is not really same as an uppercase E with the accent. So, because of that, it is not stable and it's not really automatic as well in the software. Next slide, please.

So, when it comes to internationalized domain names, as I told you earlier, to use Unicode for domain names, we have this IDNA 2008 standards. This is an IETF draft as well. The drafts are mentioned over there. So, we have to do that normalization, as I told you earlier.

Also actually, now, there are some label generation rules that could apply, especially now when you think of different scripts and so on, how we have to use these different glyphs—different characters in those scripts and so on. The language rules has to be considered when we actually define, especially now, considering the security and stability issues. Earlier, they have a lot of discussions about similarities between those different characters and phishing attacks can happen.

So, because of all these situations, the security and stability considerations has to be taken into account. And then, because of the community-based work has happened depending on the scripts that

has been used by different communities. So, there are these root zone label generation rules for top-level domains, as well as for second and other level domains as well. And those are available in the in the community-based website that we have. Next slide, please.

So, now, let's move into … And I hope that you got some understanding on the IDNs and Unicode and so on. So, now when it comes to the email address internationalization, now simply we call EAI. So, we need to actually define what is really EAI and what is really not EAI actually. So, what's EAI? If we are having the UTF support, as we discussed, for the mailbox name … So, when you consider email ID, of course, in this discussion, that's the real focus. We are talking about the email ID itself. So, in the email ID, we have the mailbox and then we also have the domain as well, which is actually … The mailbox is what's coming before the @. And then the domain name is after the @.

So, if we are talking about the UTF-8 support for that, then obviously, then we call that as an EAI. But if we are talking about, for example, things like the subject line, or if we are talking about the address commands or the message body and so on, that doesn't really come into the EAI context as well because that's all handled by the conventional mail stuff, like the MIME and so on. And also, if we are talking about, say, other encoding schemes other than UTF-8, that is also actually—do not come into the context of EAI. Next slide.

Okay. So, with these considerations we had, now we can try to categorize the domain names and email addresses into different formats. So, if you consider the domain names, now, also from a UA

perspective—from a Universal Acceptance perspective—we have the new top-level domains—the short ones, and then we also have the longer ones as well, and then internationalized domain names. So, all these things have to be accepted by the application. So, not only accepted but also processed, and validated, and so on—displayed and so on, right? So, everything has to be done and that's what we call Universal Acceptance.

Now, when we get to the email address internationalization, there are different categories that we can consider here. So, the mailbox can be an ASCII mailbox and then the domain can be IDN domain. And then the mailbox also can be the UTF-8-based mailbox and then the domain can be ASCII. So, there are some examples given here, you can see in the slide. And then, the mailbox can also be UTF-8 based and the domain can be IDN. And then also, all those type of scripts, we write from left to right, but also there are some other scripts that we write from right to left as well like, for example, Arabic. So, we need to actually cater for all these different categories of email IDs in this case. Next slide, please.

Now earlier, there were some note on the UASG, the Universal Acceptance Steering Group. Now, UASG has done quite a lot of work related to all these things. And also, actually, a number of studies and white papers have been published, and surveys have been done, and so on. So, you can see that there are—if you go to uasg.tech, there are quite a lot of documentation and these documents have certain numbers.

So, here, I'm referring to one of the documents which is UASG027. So, in this case, this is a survey that has been done based on top 1000 websites

or so globally. And so, considering the support, actually, for email addresses here. And you can see that when the email IDs are, for example, like ASCII mailbox with a new short, new TLD, the acceptance is much better. It goes to or beyond 90% or so but still less than 100%.

But when it comes to say—when the UTF-8 based mailboxes—for example, like Chinese scripts or Arabic scripts and so on, then the acceptance is pretty low, actually pretty much no less than 10% or so. But when you consider the last two to three years or so, we have seen some improvement especially like there is some improvement in 2020. And so, there is some slight improvement. And the whole point here is that actually now creates some more awareness so that more communities try to actually support these standards. Next slide please.

And also, actually, this is another survey that was done. This is basically to analyze the EAI support in email systems, the mail servers and based on the MX records considering the domains over there. So, again, here there's pretty much less than 10% of the mail servers could support that as well. So, this is based on document, again, UASG021T. Next slide, please.

So, as you could see based on those studies as well, we have to do a fair bit of work in terms of supporting EAI in our email systems. So, the whole discussion here is actually if any of you here managing your email servers, to make sure that the email servers can have the EAI support.

Now, there are different levels of EAI implementations that we can consider. First one is, of course, there is no EAI support, right? So, that

means only the ASCII email addresses can be supported by the tools and services. So, here we say there is no EAI support.

And then we also have what we call level one. So, in this case, you can receive the email from the EAI addresses and then you can also send the email to EIA addresses as well. But we can't create mailboxes and domain names in UTF-8 format. So, there are some … Say for example, if certain email service, if we can just send the emails and receive the emails only without creating our EAI based email addresses, we will consider those as level one.

Whereas level two is a level one, plus we are able to create the mailboxes and domain names in UTF-8. So that is what we call level two. So, depending on different tools and services, some of the tools could be having level one support. Some tools could be having level two support and some may not really have any EAI support as well. S,o we can have different categories in this situation. Next slide, please.

Okay. Again, this is also another kind of in a snapshot, again taken from UASG030 here, an evaluation of EAI support in email software and services. So, that's the report. So, here, you can see that in email systems, we have different components, like we have MUAs, MSAs, and mail transfer agents, and so on. So, depending on those different components and then the different tools and services, what sort of support do you have, based on the different categories that we discussed earlier, like L1 or L2?

You can see some of those [servers], they do have L2 support for different components like MTAs and so on. And then, some has got some

partial support. Some do not have. And also, actually, some of the tools say that have not conducted any testing yet so we don't have data for those. So, this is just another snapshot of available tools and services. And then, for more information, you can also refer to that report we just published in UASG030. Next slide.

Okay. Now, let's consider things that we need to understand, especially in the case of figuring for EAI. Next slide, please. Okay. Now, as we know, in terms of the email protocols, we use SMTP. And also, we use POP and IMAP. So, there are some considerations based on these protocols. We have to make sure that SMTP is also augmented to support EAI. Now, between the SMTP servers, there is some protocol negotiation has to happen. And then, there are some signaling flags that would involve in this communication. So, here, we have a certain signaling flag called SMTPUTF8. So, this is actually to specify the EAI support. So, that is quite important in this case.

And the other thing is that all the email, or all the SMTP servers in the path, they must to support EAI if they are to deliver that email successfully to the receiver. Now, there could be many SMTP servers in the path. So, we have to make sure that the email gets properly delivered. All those SMTP servers should support the EAI.

When it comes to the POP and IMAP, again it should be augmented to support EAI as well. And then, there are some signaling flags to specify for the EAI support here. Now, as we know, we use POP and IMAP in mail delivery agents. So, it is possible that we could do some sort of half support here and deliver downgraded email to the client—to the user.

That's is possible. But it is actually not really recommended because there are some issues involved here. So, it is possible but not really recommended. Okay. Next slide, please.

Okay. So, let's try to take an example here. You can see here all those different components that I spoke to you is listed for an explanation purpose here. You have a mail user agent and then we have your MTAs, just for explanation purposes. And we also have the receiver, the MUA.

If you consider the signaling happens between two mail servers, two SMTP servers, two MTAs, we have certain negotiations happen between those MTAs. So, the sender will connect and then the receiver has to actually accept and respond back with certain signaling that it can support. Say, for example, if it can support eight-bit MIME, it should respond with a certain response code, saying that it can accept, so that the sender can send those.

So, in the same way, the important thing here is that in this signaling, we need to consider this SMTPUTF8 signaling flag for the EAI support. So the receiver has to respond with the SMTPUTF8 signaling flag so that the sending MTA knows that the email can be sent or delivered. Next slide.

Yeah. In the same way of, as I told you earlier, even for IMAP and POP protocols, we need to have some signaling flags. So there is actually for the POP. We have a UTF-8 command. And then, for IMAP, also we have to enable UTF-8 accept command as well. Next slide.

Also, considering these protocol changes and some delivery path considerations, there are a few things that we have to consider. To send and receive an email with EAI, all the email parties involved in the delivery path, they have to be updated for EAI support. This is very important because the thing is that, say, for example, if you have multiple mail servers. And then, if, say, those mail servers could have different priorities as well. And then, say, depending on the priorities, the email delivery can take different paths as well. So sometimes, if you take a certain path and if all the mail servers in that path can support EAI, that is fine. The mail can get delivered.

But then, if you find, say, one of those mail servers in the path do not support—and also, actually, if you have listed as an email server with email support, then depending on the priority, if it takes that path, the email may not get delivered. So, sometimes that mail gets delivered. Sometimes, the mail may not get delivered. So, there can be always some issues in that way. So, that is why it is very important that all those mail servers have to support EAI. So, even if a single SMTP server in the path does not support EAI, then the email will not be delivered. Next slide, please.

So, what happens when one email server in the path does not support EAI? The thing is that, actually, to know that stage, obviously, the MUA will not know that beforehand. Only, it will know at the next top level. So, for example, once the MTA gets the response that the next top cannot accept the EAI email, then only the MTA will have to send—basically, drop the email and then send the user an “unable to deliver” report, basically. So, we can’t really have this understanding

beforehand, at the MUA level or the client level. So, the delivery failure message has to be delivered back to the user. Next slide, please.

Now, if the receiver client does not support the EAI, as I told you earlier, it is possible to send a downgraded version of the email to the client. That is also defined in the RFC as well, RFC6857. But this is not really recommended, mainly because there could be some issues. Say, if the receiver doesn't support the EAI, then it may not properly display the glyphs or display the characters and so on. So, there are issues like that.

And also, if you have to respond back and so on, there could also be issues, especially if it's a mail delivery agent. If it doesn't have the SMTP function, we may not be able to deliver a delivery failure message as well. So, the bottom line is that. Downgrading is possible but it is actually not recommended to do the downgrading. Next slide.

There are some additional considerations as well. The case folding is one of the things. In ASCII, they use case folding because, say, according to this example, you take, say, PETER in uppercase or peter in small case, it doesn't really matter. The mail will be delivered. But when it gets to EAI, that's not the case. It is not automatically implemented in most of the EAI-ready software.

When it comes to spam, the considerations are such that we still have to follow the usual best practices—things SPF, DKIM, and so on. We still have to follow those. But the thing is that because there are these other glyphs—the characters in the email ID, and the subject line, and so on—the spam filters could be still filtering the spam. That is possible. So, this also needs to be taken into consideration.

And then, there are software service and so on. Basically, not every server or client doesn't really support EAI. So, that's the current situation. Next slide, please.

So, there are some considerations for mailbox names. So here, we are talking the left side to the @ here, the mailbox. Let's see what are those considerations. Next slide, please. In fact, in the previous presentation also, there was some discussion on the script mixing. Now, here, we are not talking about the domain itself here. We are talking about the mailbox. So, there could be some local language, local preferences, local practices that might need to consider, depending on the script mixing.

But it is always recommended to allow some limited script mixing here, only if it's clear—if there is really a need for based on local practice. But also, it is very important to consider the security considerations, mainly because there can be some confusions. There could be some similar scripting and so on. So, this needs to be always avoided and make sure that such confusion should not happen. So, the security should be considered very importantly here.

And also, preventing invalid and unstable rendered strings. That should also be considered as well. There are some reference IDN tables available. So, we should also refer to those tables in terms of selecting what are the appropriate strings that we should use.

There is a string validation tool available—LGR tool. This is available through the ICANN Org website as well. And we can validate what are those mailbox strings. And also same with the right-to-left scripts as

well. Consider those. Especially avoid script mixing with the right-to-left scripts to avoid more confusability and security issues. Next slide.

And then, aliases and display name considerations. This is actually something recommended because due to these earlier-mentioned reasons, there are possibilities that because of those EAI issues and so on, the EAI mailboxes may not really get the email. So that is why always having some alias is recommended. So, that's something that should be considered.

And then, signs and symbols. Earlier, I was talking about ASCII aliases, right? So just to [lead here]. In terms of signs and symbols … Also, in typical ASCII-based email IDs, sometimes we use things like dot, underscore, and hyphen and so on. In the same way, even in the EAI-based emails, we should also actually use any appropriate signs and symbols from a local context—from a local point of view. But always, it is important to review any additional signs and make sure there are not any security issues, if you are using any different signs or symbols. So, that's important. Next slide.

Yeah. I think important thing, again, is to get involved. So, I have also referred to a number of documentation, especially with the UASG website, uasg.tech. There are lots of reference materials, white papers involved. Next slide, please. There are some references where you can get involved as well. There are email IDs here and then the website itself. There are some discussion email lists that you can subscribe to. And then, various UA working groups. There are a number of UA working groups as well, depending on those different … We discuss about

different components of these whole email systems, depending on that there are different activities, categories, and so on. Next slide, please.

So, all this information is available from the uasg.tech. And then, these are some references for those different documents that I was talking about. I think this pretty well will wind up my presentation. Next slide, please. Yeah. I think that's about it. So, if there are some questions and discussions we can take, I believe Sarmad and [inaudible], also, I believe they are in the room as well. So, we can have some Q&A or discussions. Thank you.

EBERHARD LISSE: Thank you very much. Not directly what we usually talk about but very interesting and comprehensive. I use mainly pure ASCII because my name doesn't have any German umlauts or anything. So, in the email address, it's easy to do. But one question that I have is you said acceptance. How do measure acceptance about Chinese emails? Do you count emails? Or can you count emails that are behind the Great Red Firewall? Or how does this work?

CHAMPIKA WIJAYATUNGA: Typically … This is actually part of this whole discussion of universal acceptance. So, we have to consider there are a number of stages here—the acceptance—as I said, the acceptance of the email ID and then the validation, the storing of that email ID, the displaying. So, there are different aspects that you have to consider here.

Now, in terms of that survey, how they have carried out different … The methodology of the survey is published over there in that UASG website, the documentation that I pointed out. And Sarmad, if you are there, maybe you can probably provide some pointers on that—how that survey was carried out.

SARMAD HUSSAIN: Sure. Thank you, Champika. And hello, everyone. That particular study, what was done was that there were different websites selected globally to see whether they accept email addresses. What we did was went to either their "contact us" page or some of the registration pages, if they were offering such a page.

And we went and used a UA address, in the case they were asking for an email contact. So, whether they accept that email address, for example, in Chinese or in Arabic. Or they used different email addresses in the different scripts to test which scripts were accepted and which scripts were not and then used that information to eventually document the level of acceptance of email addresses.

So, in a way, it was just testing just the input and initial validation of that email address and not really the complete processing of email addresses because I guess you would appreciate the complete processing would also entail that after the email address is accepted, the website can actually send an email to that email address as well. But we stopped short of that process and this was just testing whether an email address was—website was just being able to input that email address or not. Thank you.

EBERHARD LISSE: Thank you. So, I can speculate that in China, they use more Chinese email addresses. We have a question in the Q&A. And I can read it, from Yoshiro Yoneya, "Have you got UASG documents to start NOGs?"

CHAMPIKA WIJAYATUNGA: Yeah. Actually, I believe you're referring to the network operators' groups. Yes. In fact, during the whole of last year, actually, we have conducted number of sessions highlighting the universal acceptance—also, actually, the IDNs and especially the email address internationalization, especially focusing on the network operators who are pretty much dealing with email servers and so on, in terms of supporting EAI. So, this messaging has been conveyed to a number of network operator groups and also to RIR communities as well. So, just to answer your question, yeah. That has been done. And we'll continue to do so as well. Thank you.

EBERHARD LISSE: Okey-doke. Thank you very much. There is one hand. I've asked this, for participants to put it in the Q&A, which hasn't happened. So, basically, I need to do one more thing before Jaromir will close the proceedings. We [quizzed] you so I need to give you back the feedback. The overwhelming majority of the participants during the breaks are in favor of keeping the breaks. So, we will do so in the future. Whether I like it or not, it's a community-based thing so we do what the

community wants. And we will, then, in the future, any virtual Tech Day exactly like this. Okay, Jaromir, thank you very much for having been volunteered by Andre to do this for us. You have the floor.

JAROMIR TALIR: Thank you, Eberhard.

EBERHARD LISSE: There you go. That's what wanted. I wanted her to bring the presentation up as a memoir for you.

JAROMIR TALIR: Great. Thank you. Greetings from Czech Republic. So, we had eight interesting presentations this tech day. I will try to give you a short summary about each of them.

So, we started with the presentation from Giovane from SIDN about the topic of how collecting of DNS TCP traffic is quite valuable because there are some interesting information in there, in particular the RTT measurements. So, it's possible to measure the latency of your requests heading to your Anycast network, for example. It's quite interesting method. We are using the same in CZ for some time and trying to distribute Anycast extensions—extend Anycast nodes based on the results of this analysis.

And the recent thing that SIDN actually did, was that they released from the cage their anteater—not the animal but the software that they've created. It's a software built on top of their ENTRADA platform. So, if

you are using the ENTRADA, probably this will be the easy extension to give you the almost real-time monitoring of your network. You can look into the dashboard to see the latency of your request. Based on this, they have even revealed some bugs in some [inaudible] that they could fix and reduce the latency of the whole network. So, it was definitely interesting. And the best thing is that it's open source so anybody can use it.

The next presentation was from Maciej from Université Grenoble Alpes. And this was about a system that registries or registrars could potentially use for detecting the malicious websites or domains. This presentation focused on the distinction between malicious versus compromised domains, since the subsystems ignore this. And ignoring that could cost some collateral damage by somebody may decide to disable some domains that only some part of this is causing an issue.

So, the system is called COMAR. And it actually includes the machine learning technology. And it's based on publicly-available datasets. So it should be affordable for anyone. This topic is particularly interesting for, maybe, EU countries, with all these discussions about regulations, and particularly NIS2 regulations, where other things are discussed at the moment. That was quite a lengthy discussion—a lot of questions on this topic. So, it seems to be interesting for anyone. There is call for open-sourcing that tool. So, hopefully this will happen in the future.

The next presentation from Ulrich from Internetstiftelsen, my fellow co-chair of the CENTR Technical Working Group. Ulrich talked about the transition of NU top-level domain from NSEC3 to NSEC. They are

managing .SE as a domain for their company. That's already running NSEC. And now, also the NU is running NSEC. So, this was like cleaning the things. Of course, both NU and .SE are public so there's no reason to hide the content via using NSEC3. And he also mentioned that there was some errors in the software, some time ago, that probably was related to NSEC3. So, the transition was like reducing the possibility of problems.

The transition is well-described in the RFC. So, they just followed that steps that are described. They created a test batch. It looks like everything well except, if I understand correctly, some glitches in configuration of Atlas probes. But now, the process is completed and there was no issue related to that. So, this is maybe the guide for some other TLDs that would like to go this way. They could use their experience. Actually, most ccTLDs is going public. You could see that .EE, Estonia Registry, and recently, .CH also published their zones. So maybe some more of them will come and this work could be interesting for them.

Next presentation was from Mark from Netistrar. Last time, if you recall, we had the representation from the RDAP solution from .AT, from the registry perspective. And here, we had a presentation of the RDAP implementation from the registrar perspective. So, a different perspective of the same things. Mark mentioned some issues they had, which is like the vCard/jCard issue that is discussed also at the [inaudible] for us, like IETF, that this probably could be somehow addressed. And on the positive side, he mentioned that they could take advantage of the network protection for services like DDOS protection,

rate limiting, or caching. So, these advantages of the new protocol was very good for them.

So, the RDAP is apparently on the rise. The sad thing, maybe, is that when the RDAP started, there were the first, actually, TLDs that are in the ICANN Bootstrap Registry were ccTLDs, like Brazilian TLD and Czech TLD. Now, there's more than 1,200 gTLDs and actually only 17 ccTLDs. So, at this point, probably, the gTLDs took the lead. And there is some work that ccTLDs probably need to do to follow. So, we had that RDAP workshop at last CENTR Technical Working Group. And several other ccTLDs mentioned that they already are implementing the RDAP so hopefully this will change in the future. But it looks like RDAP is finally taking off so that's good.

The next presentation was from Benno from NLnet labs. It was a little bit about the DNSSEC cooking. He mentioned there two steps for DNSSEC Key Ceremony, where you actually start with a [inaudible] recipe and then you cook it—you actually execute it. This is closely related to the scenarios where you have the [offline] KSK. That means that you separate the roles of the teams. One team is actually doing the ZSK administration and the other one was KSK. So, the one team will prepare the recipe, which is actually how the zone will look like in the next half a year, from example. And then other party will cook it and assign this list of DNSKey resource records, which can be then put into the zone file.

And this is already probably implemented in OpenDNSSEC, the tool that [inaudible] provide. It's not DNS but there is a similar possibly with

a little bit different names and different formats but should be quite the same. If you are not familiar with the ICANN terminology, then the recipe means the key signing request, KSR. And the cooked recipe is the signed key response.

So, the next presentation was from Iliya from Edoms. And he walked through some ccTLD security practices. He described various registry components and interfaces and some hints how to secure it. Maybe those things are not new for big registries with operating under ISO27K certification. But it was a quite interesting checklist for small registries that—what you should check to not to forget about when doing the security analysis of your system.

Next presentation from two Brians, one from MarkMonitor and one from Donuts, was about homoglyphs and homographs in domain names. They described that there still are cases where it is possible to play with the names of the domain to, for example, prepare a phishing site. And there are some policies by ICANN, like that it's not possible to mix scripts in the labels, in the domain name. The issue is there are those scripts that have large character sets, trying to include more languages.

So, they also mentioned some ways how to get over that, like, I think, a language track tool, registration as a possible migration, or maybe to block the possible confusable characters after the registration of the ASCII domain, based on the domain. So, at the end, they mentioned that probably, there is just a low number of cases on the current

phishing scene, based on some resources. But they still need to be investigated a little bit deeper.

Also, they had the discussion, phishing is probably still an issue. We have an interesting similar situation here in Czech Republic at the moment because we are heavily discussing pros and cons of different methods of electronic identification and that the people are—that the phishing actually can be used to steal credentials at some two-factor authentication methods, like the OTP and things like this. We are promoting using the phishing-resistant standards, like FIDO, for e-government. So, this is actually a topic that we also mentioned quite heavily in the recent discussions.

And the last presentation was from Champika. It was an overview of things related to email address internationalization. So, it covered all the descriptions of an issue of IDN and EAI—what is it and how to support better acceptance of these technologies. Actually, if somebody hasn't understood some of the previous presentations, probably the presentation explained them what is it about. And the outcome, actually, that looks like the numbers are still bad. He mentioned that only 10% of email systems support full Arabic email addresses, which is a little bit scary. And hopefully, this will get better soon.

So, to conclude, they gave us a lot of good topics. Probably, DNS abuse is still on the spot because of the questions and discussions that we had today. Probably only disappointing issue about this today is that we still cannot meet face-to-face and maybe continue this discussion somewhere at Cancun beach with a glass of tequila. So, we still hope

that this is going to change soon. I will hand over back to Eberhard to close the meeting.

EBERHARD LISSE: Thank you very much. We had, at one stage, 168 combined attendance. If we deduct staff from it, it's 106 panelists and attendants, which I think is quite good. I am looking forward to do this again in June. So, I must say, if they let me, I will probably fly to Europe because I've got some other stuff to do. And I will do it then. I will have the meeting, then, from Berlin. I will be there in Germany, probably. Otherwise, we'll do it from here. It's going to be the same.

Thank you very much, everybody. Thank you very much, Jaromir, for stepping in on short notice to give us your view on things, which is important. And I am closing Tech Day now. And fourth block that we had available, we have given back. And I don't close without thanking the ICANN MTS technical staff for running the meeting and our three cohosts, Kathy, Kim, and Claudia. Thank you very much for your usual efficiency and your running the meeting. Goodbye.

KIMBERLY CARLSON: Thank you all. Bye.

[END OF TRANSCRIPTION]