SAN JUAN – Tech Day, Part 3
Monday, March 12, 2018 – 15:15 to 16:45 AST
ICANN61 | San Juan, Puerto Rico

UNKNOWN SPEAKER: Tech Day, Part 3. Monday March 12th, 2018. From 3:15 to 4:45 in the evening, Monday March 12, 2018. [AUDIO BREAK]

EBERHARD LISSE Can we come and settle down again in the back please? [AUDIO BREAK]

Okie doke, so thank you very much for coming back after our short little break. On the agenda, it's Don Hollander to speak, to give a follow up about Universal Acceptance, but it's going to be Mark Svancarek from Microsoft.

MARK SVANCAREK: That was a great introduction, thank you, thank you, thank you, thank you.

EBERHARD LISSE: So unintentional.

MARK SVANCAREK: So, hi everybody, I'm here to talk about Universal acceptance. First, we'll tell you What is It, we'll tell you about our Steering group which is active here at ICANN. Some challenges, then we'll talk about some bidirectional stuff which is one of the challenges, so it says challenges, then there's another challenge, and then finally there is no conclusion except just, you know, join us and do it.

So, the definition of Universal Acceptance is; that all domain names and all e-mail addresses should work in all Internet enabled applications, devices, and systems. And we all know that there are many challenges. It's better than it used to be, but all domain names and all e-mail addresses currently do not work in all Internet enabled applications, devices, and systems.

So, to address this, the Universal Acceptance Steering Group was created, we are a community-based team that is supported and funded by ICANN, and we try to identify the top line issues and propose solutions and share best practices. Our objective is to help people who write software, and people who maintained websites, those sorts of people to keep pace with the evolving standards. We're very much about standards and having people adopt the latest standards, and then our message is that there is an imperative here; it's very important to support Universal Acceptance because it's an enabling technology, a collection of

enabling technologies that allow people all around the world, including perhaps the next billion people to build their online spaces and maintain their online identities.

And I'm just going to go quickly through some of these slides, because we do have a website as it shows there; usg.tech and a lot of great material is up there, so feel free to do that in your spare time. But, our activities, as you can see, run a gamut, so we're reviewing a lot of things; we review popular websites, we look at development frameworks and open-source tools. We review browsers, and operating systems and e-mail systems, and we build test cases and test environments and we built communities. And we have outreach, live workshops, we give them here at ICANN, we've done things at IGF, we go to other countries and discuss things with communities there, and then we've been writing a lot of things, knowledge basis technical documents and all sorts.

So, I mentioned there were challenges. If there were no challenges, we would not need a Steering Group. And, the challenges of course are as always, there's technical challenges, and there's business challenges. So, the technical challenges include things like challenging old assumptions, updating old software, managing backwards compatibility, and that's the sort of thing I'd like to talk about today, both in my presentation and

then in the questions. There are of course business challenges as well; convincing people that there is a business opportunity to justify doing the work and understanding as part of that opportunity what is the return on the investment. We shouldn't really talk about that today; this is tech day, but we have lots and lots of material up at usg.tech that can explain more about this, about what are the opportunities, and materials to help you convince other people to look harder at those opportunities.

So, technical challenges; sometimes people just make bad assumptions. Bad assumptions as their old assumptions, let's say. Old assumptions about domain name strings and e-mail addresses. Sometimes that's because things change, for example; SMTPUTF8, there used to be SMPT, then there was SMTPUTF8. If you weren't aware of that change, that there was a new thing, you might not have gotten around to updating your software. Sometimes standards are misleading; if you are following HDML 5.3 you may know that they had added some input type definitions, and the one for e-email was actually wrong; it didn't work for internationalized email. It was just simply wrong, and it took a while to get that spec updated, so first we had to find out that it was wrong, and then we had to engage and now it's in the process of being changed. So, if you had followed that standard, if you had re-written your website to use that input type, you would have been wrong.

And then, in some cases, such as linkification, and I'll talk about linkification later, there may not be really a spec at all. There's just little bits of, you know, trade secrets and private knowledge that various software providers have implemented in their products, all of them doing it differently; perhaps not even doing it consistently from one product to another within the same product line.

But honestly, most of the assumptions are based on, people didn't look at a document at all; they just sort of had a heuristic sense of how the eco system was and they coded around it, which is why sometimes you still find hard-coded on a website a regex that says, "All domain names, all TLD's are 4 letters or less."

So, there's an example right there; all TLD's are 4 letters or less, or maybe it's 4 and then 6 or something like that. Or, the string can be only so many characters, so the document is very clear; 254 I think characters, each label can be of a certain duration, or perhaps someone is maintaining a list, a white list, which has not been compared to in an authoritative source for some time. We found that when the new gTLDs were coming out that the browser manufacturers had outdated methods of keeping up to date, and they were making assumptions within the browsers.

So, there are bad assumptions about that, e-mail addresses, there are similar assumptions of course because they contain a domain name part. I'd mentioned that HTML 5.3 has an e-mail input type definition which was wrong, but usually people use regex's on websites to determine if something is an e-mail address, and I think these days it's probably just safest to say, "Does it have an add and a dot in in it," and just take it as an input string and go and validate it by sending an e-mail to that address. But, people try to get clever and they try to write a regular expression. I think you all know the joke, "I had a problem, so I decided to use a regular expression, then I had two problems," and that's definitely the case related to universal acceptance.

And then, there's things related to this; if you're really overthinking what an e-mail address should look like, or what a domain name should look like, that can show up in other places; spam filtering is an example of that too, and then it can get really esoteric like, I see multiple scripts in this identifier, so I'm going to start throwing up warnings at the user to scare them off; this looks scary, this looks weird, and it's only weird because you haven't read the spec or you're not familiar with the spec, because your knowledge of what is weird or not weird was based on some previous version of the eco system.

And then finally, linkification. Linkification is the idea, and I'll define it in another slide but I'll say it now, is the idea that sometimes when you input a string into your e-mail program, or a word processor or something, a text box or a chat window, that the software can guess the users intent and say, "That looks like it should be an e-mail address, and make a link with the mail-to protocol," or, "That looks like it should be a domain name," and add http, and make it a link like that. But, you have to understand the user intent, and you have to understand it consistently, or else you have no hope of really doing this in an effective or consistent way.

So, here's our usual example of; I've entered a perfectly valid e-mail address, and without even attempting to send to it, the software says, "You know, that just doesn't look right, why don't you use a real one?" And of course, this is repeated 100 times a day, I would like to say 1000 times a day but we're still in the early stages of EAI, 100 times a day people enter in e-mail addresses that either use Unicode characters, or maybe they just use the new gTLDs you know? Like .realestate, .realestate has a lot of problems because of bad assumptions in online software about what TLD's should look like.

So, we like to encourage people to update their old software, and in our experience, a lot of it is just a bug fix; something

pretty simple. I mean, we can't make the case that it's always pretty simple, but usually it's more along the lines of a bug fix. You know, something that you file, and your team goes off and does it, and you don't have to do any budgeting for it. Sometimes it could be more significant, more like a design change, where it has a project has to be created around it, but you should usually assume that this is something relatively small you know, one or two lines of code, easily tested, something you can do quickly and without much risk.

The tricky parts are actually finding all the places that you have to update. So, some of them can be really, really obvious like contact information in a CRM database. Okay, those are e-mail addresses, I should probably check how those work. Domain names in a browser; okay, that's completely obvious. But sometimes they are not; sometimes API's may be moving around URI's or in log files, or error logs, or all sorts of places in your software, and so, you have to figure out; how do I know that I've actually touched them all? How do I know that I find them? So that could be tricky sometimes.

And then, the corresponding use cases; how do I actually exercise that part of the code to make sure I've fixed it correctly?

There are also by directionality now that we're talking about Unicode here, there are some languages that are bidirectional,

and if you're not accustomed to using bidirectional scripts, that can be kind of daunting and confusing, and then linkification, as I mentioned earlier, since it's not really spec'd anywhere, everyone has a different opinion about how to do it, which means everyone has a different opinion about when it's done correctly.

And I know, I said I wouldn't talk about business challenges, but we do have to admit that once you're talking about here's how you update old software, you have to acknowledge that if there's no obvious return on investment, people generally don't want to do it.

So, here's an example of the last set of challenges, so I mentioned it was; challenging old assumptions, updating old software, maintaining backwards compatibility, and so in this case, I'm going to talk about EAI; E-mail Address Internationalization. So, this is a case where we designed something that wasn't backwards compatible, so IPV4, IPV6, not backwards compatible. SMTP, SMTPUTF8, not backwards compatible. And they tried, they tried really hard, and they always kept finding cases where it just didn't work.

So, you can see in this picture, conceptually there are two e-mail systems running in parallel; there is the legacy system, and the legacy system can input messages into the new system, and

there's the new SMTPUTF8 which cannot communicate back into the legacy system without some sort of downgrading. And, the problem is, we've tried to find downgrading solutions that always work, and really in general they don't. There's one specific case which we've documented, that works in very specific circumstances, but in general when we talk about downgrading it's; don't.

So, here's the one that we support, we invented a name for it, we call it, "Downgrading With Aliasing," and that means if you are an e-mail provider like Xgen Plus, Ajay over here, or Coremail, you can assign your user another e-mail alias, and then you can determine on the fly which address to use. In any other case, you can't do translations; you can't transform an address from one form to another form, because somewhere it's going to break, even if you think that it's not going to break, it probably will.

I don't know, I've got a rendering problem here; it says other transformations are—

EBERHARD LISSE: Just go back and forth—

MARK SVANCAREK: Oh, is that how you do it?

EBERHARD LISSE: Alright, it doesn't work.

MARK SVANCAREK: I don't know, forbidden I think is probably the word, or you know, disallowed, or not recommended. So, in the example that I just gave about downgrading with Aliasing, the assumption is that you own the mailbox; you are managing the mailbox that is being transformed, and so therefore you have knowledge about these two Aliases go together; I am UTF8 enabled, the destination isn't, but I have the knowledge of the mailbox and the Aliases, and I can choose which one to use.

In all the other cases where you don't have that information, you can't do the transformation. And, an example that comes up all the time, at least once a month on our chat rooms, on our message lists, in my engineering groups, somebody learns about Punycode and says, "What about Punycode," and then you have to have the same conversation again and so we always say, "Don't transform things into Punycode in the local part."

Punycode is designed for the domain name part, and if you do receive an ace-encoded, a Punycoded local part, don't get clever

and try to transform it yourself back into Unicode. They'll say, "Well, isn't that safe, are you sure it's not safe?" So, I have an example here, I lost my connector—here we go, okay. So, I'm sure everybody here is a bit nerd and knows who Son Goku is, right? Of course you do. He is a very famous character from Japanese Manga and he's based on the Monkey King of Chinese legend.

Anyway, that's his name there, Son Goku, and suppose I want to have that mailbox, and I say, "Aha! I notice that there's an ace encoding of Son Goku, which is, XN- - 98SY4JMV0A. Cool!" So, if my friends don't have EAI, they can just use this, right? They can just use that address and send it to me and it'll work, right? And the answer is; no. Because, in fact, I, me personally, already own that string on Outlook.com, that ascii string; you can't make an assumption that that thing doesn't exist and that a name collision won't happen, and this is as unlikely as it may be, it may seem very unlikely that that name exists, you can't guarantee that it doesn't, and every time you go down the downgrading path, you will inevitably come up with some case where it's like, "Well, that seems very unlikely," but you can't count it out.

John Lavigne and John Clensen and Barry, I guess they actually polled the eco system, and they found examples out in the wild

of pretty much all of the hedge cases, that nothing was truly an edge case; there was somebody running their own server who had instantiated each of these examples, and so in frustration they said, "No downgrading." So, in UASG, because we are very stubborn and foolish people, we said, "Well, yeah that's great guys, but we're going to look at it ourselves," and we came to the same conclusion. So, no, there's one kind of downgrading that we recommend, we call it this, and it only applies if you're running your entire—if you know everything about that mailbox. So, if you don't own that mailbox, you cannot transform it.

And that means, going back again, effectively, there are two e-mail stings; there's an SMTPUTF8 e-mail stream, there is a Legacy e-mail stream, and over time, as everyone gets upgraded, all the old mail will just flow into the new one, and it will seem like it was never a problem; you never know how long that will take though.

So, what are we doing about this? Lots of companies have been working on this for awhile, we've been building a community around this. I'm from Microsoft so I'll just put in my little ad from Microsoft that we recently announced support for EAI in Outlook, Office 365, and Outlook.com, so we're supporting what we call, "Phase 1," now. Phase 1 is the ability to send to, and receive from these addresses, and so you can see there is a

number of solutions that do it. Gmail does it, Office 365, Postfix, XM, Halon. We haven't verified all of these; that's why it says, "They claim support," but what we're doing is we're building an evaluation program.

So, on a previous slide I showed UASG does a lot of things; we create outreach, we teach people, but we do a lot of evaluations and then we post those results online. So, we are about to start doing EAI evaluation and we are looking at what we call 2 Phases of achievement; one is, can you send these address, can you receive from them, can you reply if you receive something?

And then Phase 2, which is actually hosting these mailboxes on your service. So, as I mentioned, Xgen Plus, Coremail, they're running services now, there are other people who have implemented solutions that we have not necessarily evaluated, but we've at least talked to them, so they'll be part of our evaluation program too.

But now I'm sort of edging into the other conversation; here's what an e-mail address looks like, here's what Universal Acceptance looks like for e-mail addresses. So, at the bottom, you can see a couple of examples where we have ascii, ascii, Unicode, okay, so that's new. Or, ascii@unicode.ascii. That doesn't work everywhere either; the one that suprises people is

localpart@alongnewthing plus alongnewthing. Like I said, realestate.lawyer; those don't always work.

But, up above, we have bidirectional. Usually, the local part is on the left, the domain name part is on the right, the TLD is at the far right, and that's how it goes out on the wire too, so on the wire U proceeds, S proceeds, E proceeds R. If you are in a right to left script, and so we have some Arabic over there, you see that things start on the right and move to the left, and conceptually this is the same; the user name goes on the wire first, then the @, then the domain, then APP. So, like I just said though, APP, wait a minute; we were going from right to left, isn't it PPA? What happened?

So, if you look at all of these things, like I've got Unicode and I've got bidirectionality, and now I'm going to try and make some links; I'm going to try and guess what a link should look like. That's pretty interesting. And, one of the reasons it's interesting is that Unicode defines an algorithm called, "The Unicode Bi-Di Algorithm," and UBA is useful, general-purpose, it's standard, and it's really about, you know, rendering script on a screen or something like that; it's really not applicable to a lot of scenarios such as creating IRL's or doing mathematical computations, things like that.

And so, I've got a link there at the bottom to the J-script page where you can enter in strings, and it will show you; this is what it looks like in a right to left context, this is what it looks like in a left to right context. And by the context, okay so…going back again, this picture here is still in a left to right context, even though there's one of them that is bidirectional, it's still assuming that this is being read by a person whose primary mode is left to right instead of right to left.

So, let's see, yeah. So, as you know, there are some languages, primarily Hebrew and Arabic that are normally displayed right to left, but, you can mix in left to right characters in there at will, borrow words and things like that, and the digits are usually displayed left to right, except for, as far as I know, a single language family called Igbo, which is from, and I hope I pronounced that right, which is a Western African Family of languages and scripts.

And then neutral characters sort of inherit, so the algorithm specifies the classifications and figures out what their visual output is going to be, and I note at the bottom that an IRI, so a URL is a universal resource locator, an IRL is just the internationalized version of that, and that's a specific case of a URI or an IRI, which are general purpose location, so like a filed

name path on a network would be an IRI or a URI. So, IRI's use schemes like http, which will maintain their left to rightness.

So, linkification; so, everything is coming together now, linkification, we actually have written a quick guide to linkification, it's on our UASG website. It's UASG 10. And, we just defined, modern software sometimes tries to create a hyperlink by when a user is inputting a string, if it looks like a web address or an e-mail name or a network path, and the example is; I typed in those letters and I got a link, http://www.icann.org so, even though I didn't put in the http, there's an assumption of user intent because they saw the www.; the software saw the input of that, so it may display it like this, or it may display it as just www.icann.org, but behind the link the protocol has been attached, so you click on it and it works.

So, the application accepted a string, demandingly determined what it should do, and that's based on some assumptions again, of what was the user trying to do in the first place? And, sometimes there are little cheaters, like if you see a protocol, if you see mail-to, say, aha! They're trying to create an e-mail address. If they include www, okay, that's probably a website. But sometimes there's not a lot of breadcrumbs for you to follow, so you have to kind of guess from greater context.

So, okay, it didn't seem like that was a problem, right? We know how bidirectionality works, we know how linkification works, why is this a problem? So, remember again, I talked about what it looks like on the wire. So, the way it's laid in memory, or the way that it goes out on the wire, which we call logical order, and sorry for that footnote there, that actually goes to my own notes in the notes thing, so disregard that one.

The logical order; so, if I'm looking at something that has these logical orders, if I'm in a left to right context, so where it says LTR paragraph mark or a right to left context, they'll be displayed a little differently, and again I'm getting that exclusion on the screen, so I think you can't see what it is. And, it's not always that helpful if you apply the algorithm as you can see in the second example on the right. The http wound up in the middle, and that's because based on the way the string gets parsed by the algorithm, it says, "Well, this was right to left, this was left to right, this goes before this," and you wind up with this unreadable thing, whereas we think that the readable order ought to be what's on the bottom there. So, heuristically, how do you resolve the problem that we have an algorithm that produces these sort of unusable results?

So, the algorithm has all these rules, it's not a giant number of rules, it's enough. But, let's think about this in a different way.

So, suppose I want to express, now this is just a string, this isn't a domain name, right? But, here's a string, and let's pretend that where it says "address," A-D-D-R-E-S-S, let's pretend that that's actually Arabic text. So, for illustration purposes right now, since most of us I presume can't actually read Arabic characters or understand the way that they transform when they're joined together. We'll just use these ascii characters for now.

So, the right to left user looks at it this way, they say, "Okay, well I have my Arabic characters, and A is the most significant character, so it goes the farthest to the right. Then the D, D, R, E, S, S, and then I've got a chunk of left to right text, we'll I'm not going to change that; that's left to right, I'm going to scan it all as a single block of left to right, and I put it like this." So, that's what it looks like. Now, by the algorithm, that's not what you're actually going to get because it gets confused by that dot, and so everything gets moved around.

So, this is how we think you should do it; so, here I've got a domain name, starting at the beginning of the domain name, again, in logical order, so I start at that protocol, and I start to scan across and say, "This is all left to right, left to right, left to right, okay I get to the separator, and now I've got a bunch of Arabic characters which are right to left, and so I will display them this way, so the second line, within the left to right context,

or the one below, in the right to left context." And this, I don't know, it's still not entirely satisfying, because you've got the http in the middle.

So, this is again the reality, as opposed to the heuristic that we would propose. Because when we look at these separators, and then we would resolve it in this way so that the thing that shows up on the bottom; this seems like a logically readable thing, so http is a protocol , it remains in left to right, then I put the separators, then I put the next chunk of left to rights, now on the actual wire it will be MSN going out.

In this context, it looks like that's backwards but since it's a block of right to left, it stays that way. But then the right to left, the Arabic.as or Arabic.sa, gets flipped around as the user would expect. So, this looks like something that is natural to the right to left user, but also makes sense to a left to right user, because they can understand the logic to it, whereas the other example, what comes out of the algorithm, really isn't logical for anybody, it doesn't really make sense.

And so, in thinking about our heuristics, you can just think about fields, or labels if you prefer, depending on your context. If we're talking about just text strings, or if we're talking about domain names. Each field maintains a consistent order, we look for leading characters that can indication directionality, we look for

paragraph markers and separators, and then we try to define, derive, what was the user intent, what was the user context, which is why this is applicable to that other problem of linkification.

So, you're trying to understand; what is the user really trying to do here? They're not trying to just blob out a bunch of characters, they're trying to express something. In this case, a domain name or an e-mail address, so it doesn't make sense if a protocol is in the middle of the string.

I think I'm at the end of my presentation content, so we can start to ask questions. This is my last slide, which is further advertisement about Universal Acceptance, and the Universal Acceptance Steering group. As I mentioned before, we've got a ton of information, so there's documents on our website, if you'd like to learn more, e-mail us at info@. You can subscribe to our mailing lists, we have more than one. You can report problems, and this is important; you will encounter websites and software that are not Universal Acceptance Ready.

You know, I try to type in my.realestate e-mail address, or my Samoan e-mail address, or my Hindi e-mail address, and the website wouldn't take it. Report them there. You can check out your own website, ICANN working with an open source provider has developed a WebCrawler so you can evaluate your own

readiness. There are people trying to define e-mail address regex's, which I still kind of feel is a futility, but people are trying to do it, and there's a project underway at IETF, so click on that link and you can see where that's at.

And, if this all seems daunting to you, go to our website and just start with the quick guides. We have quick guides on most of the topics, they are very short and to the point, and you can get a sense of each problem area, UA, all up, linkification, e-mail addresses, things like that, and decide if you want to dig in more. So, that's my presentation, are there any questions?

EBERHARD LISSE: I'm taking one question, because we are running a little behind.

MARK SVANCAREK: Am I? Sorry.

EBERHARD LISSE: No, no, that's fine.

RESPONDENT: Question/comment; I was just using a Mac and received an EAI e-mail from somebody at the conference, I couldn't reply to it on Apple's mail.app, it's completely broken, but you can add Mac to

the list, Mac works nicely. So yes, there's a lot more software made to make work, and—

MARK SVANCAREK: Even within our own offerings, Mac Office isn't due out until this summer, and there are a variety of reasons for that, not the least of which is that they changed their networking stack, and those bugs were higher priority than finishing EAI, but even we, even though we've announced our support, we do have e-mail on the Mac is entirely working and active sync on the PC is entirely working.

RESPONDENT: Awesome, but Apple did fix Terminal, it had a lot of trouble with left to right text and now it's a little better.

MARK SVANCAREK: Good, thanks.

EBERHARD LISSE: Thank you very much, a very complicated issue and I was really wondering, I speak what now, about 2 and a half languages, and this looks like…yeah. Obviously, a remote question?

RESPONDENT: There is a remote online question asking; does Universal Acceptance regulate usage of Unicode within the URL, since Unicode can be used in a cyberattack?

MARK SVANCAREK: No, Universal Acceptance Steering Group, or the concept of UA does not regulate those things, those are regulated in other places. So, there is the IDNA standard, which is referenced in our documents, there are legal generation rules and there are other contractual restrictions that apply to different domain names at different levels historically, so we simply say, "You should be aware of the standards, and the requirements, and you should utilize the latest versions of them," but we do not have any sort of a regulatory role, we can only recommend what we determine to be best practices, and recommend against what we determine to be counterproductive practices.

EBERHARD LISSE: Okay, thank you very much. Give him a hand please. And now Jaromar Talir from Seasonic will give us an update about FRED, the other open source registry software.

JAROMIR TALIR: Thank you, Eberhard, good afternoon. My name is Jaromir, I'm from CZ.NIC, and I would like to give you a brief update about what we have done over the last year in our software; FRED. For those who may not know this system, it's a set of open source tools used to run a domain registry. It has all the features that you would expect such kind of software has, like EPP interface, it provides support for DNSSEC, it has Whois interface or RDAP interface, and new WHOIS protocol. To follow it a little bit about the previous presentation, that it also allows Internalized domain names to be registered there as a TLD, as a second-level, as name servers, and I think the internationalized characters can be used also as e-mails in addresses, so I would say that FRED is Universal Acceptance ready, if I can say it.

The system is developed by Seasonic since 2006 and we are using that successfully over those 12 years. Last week we have released a new version; 2.35. It's not yet on the website, it's quite recent, but I will do the update of the website soon. On this slide you can see the link to the website where you can find all the information. At the beginning of last year, we have redesigned the website a little bit, and one of the most visible features of the new website is the map of all the deployments, where the FRED is used right now.

As you can see, those countries are, or registries are those that told us that they are using FRED and they didn't tell us that they stopped doing that, so I believe that all of them are using it right now. Over the last year, we had 2 new small countries that started to use FRED, Lesotho; a small country in Africa, and also the first Asian registry, Macau. Macau is actually using FRED not just for .mol Latina, but also for the Chinese Macau characters as well. Here in this region we have two countries, Argentina and Costa Rica.

Costa Rica has started to use FRED like almost since the beginning, for quite a long time ago. Argentina in 2016 I think, and it's the second biggest country using Fred with their I think more than half a million domains, so after the [inaudible] of course. And I think there are even more countries right now in some stage of deployment of FRED here in Latin America as well.

During last year, or over the previous years, we have always provided binary packages for LTS versions of Ubuntu because we are running Ubuntu and we are providing also the Fedora packages and last year we added also packages for APAL, so you can use FRED even when you are using the price Linux incentives. I know there is at least one registry that is doing that, so they have to build it for themselves, so now we are providing packages for it.

There is sort of a popular tool among the registrars called WHMS and we have been asked always whether there is some plaquing to this popular tool to connect to registrar systems to FRED and it's a commercial tool, we don't provide these packages for these tools but a guy from Kenya, Michael Musia [ph.] actually wrote a plug in and made it open source, so if you are using FRED and your registrars are using the system, they could use it as well. I haven't tested myself, so I trust it's okay, definitively would need some more testing to prove that it's usable, but it's open source; anybody can take it and modify it or fix the box.

Over the last year we played a lot with documentation, we have actually hired a technical writer, a girl that's started to work on the documentation from scratch, and she built a lot of new documents over the last year, particularly most interesting I would say is the APP reference guide, where you can see all the comments to APP with descriptions, parameters that we were told by our registrants that this is really useful, they can just go there and see all the details about all comments.

The documentation is on the Get Up, and recently we have a long sort of survey on the mailing list among the people that are interested in FRED, what they would like us to concentrate on in documentation first. We have a few feedbacks from this, but if you still think that maybe you would like to particularly see

some details about the software, then just go to the link on the slides, and fill the survey.

The most part of last year we have spent on working on an automated DNSSEC, this is something that my colleague Andre presented at the last ICANN DNSSEC workshop, so I will not spend too much time here about this, just I will refer you to go to the slides from the last ICANN meeting. It's actually about the registry taking responsibility for managing DS records in the registry based on the signal from the DNS operator where by presenting specific records called CDNS records they can signal that they would like to have DNSSEC set up in the registry, and we are scanning the whole zone file for these records and trying to automate the DNSSEC as much as possible.

To make it useful, you also need DNSSEC signer that provides these records, and since we are also writing DNS software called Knot DNS, which provides automated DNS signing, so we build this feature into this software, so if you take these two things, that you can have perfect DNSSEC automation in your registry. And, as far as I know that even near Costa Rica who is using our system is recently in the last phase of preparation for deployment of the system, so there will be not just us using this for DNSSEC automation.

Last year also we completely re-wrote the web WHOIS that was the really old part of code based on the template engine called [inaudible] so we rewrote it to be as a regular jungle application. All of our software -- part of the software written in Python is using jungle application, so it's now much easier to be integrated into another jungle project, and it has a much cleaner structure of template files so it's much easier to customize this new WHOIS.

One particularly new feature of new WHOIS is something that we were asked for by our committee. Sometimes if they go to the website and print WHOIS website and they want to go to some court or somewhere with the information, they would like to have some validity in this output, so what we provided was that a user can generate a pdf with the WHOIS output which is signed by our key, secured by our certificate so at least there is some better credibility on this output that they can use when communicating through law enforcement agencies for example.

We also updated our RDAP, RDAP is new WHOIS protocol, we migrated to a new version of jungle, the stable version jungle, did some configuration, clean up. I just want to mention here that recently Costa Rica also deployed the RDAP, so if you're look into current IANA tables for all the registries that have already deployed RDAP, you could see that Argentina and Costa

Rica is there because of FRED. Or, you can look at, another way, because there is also .br, Brazil that Argentina, Brazil and Costa Rica is half of the whole RDAP deployment that's taken place here in Latin America which is great.

We concentrated a lot about refactoring, because there was some kind of, especially in the back-end part of the FRED, which is written in C++. There was a code that we haven't touched for many years, since 2006, so we did a complete rewriting of this back end. We migrated all the C++ to a new C++14 standard and updated distinct framework which I suppose should lead us to the way that we will be able to have much faster implementing of new features, so it's a wish I would say.

So, one of those features that we have implemented during last year is the implementation of postal address in EPP. Actually, in our registry the address has always had two meanings; one meaning for identification, as a permanent residency address that you have on your passport or ID card, and it's for communication because we are truly sending some snail mail letters as a last resort communication when the domain is going to expire. So, many times those are two different addresses to people, you surely don't live on the same address that they have in ID card, and we have implemented this distinction in the data model in the background of the system for many years ago

because of our identity service, but now since last year we also allow our registrars to separate these addresses and provide this postal address via EPP extension IPP protocol.

The other feature that we implemented recently is about the mail archive, because we are trying to store all the e-mail communication that FRED generates and sends for all internal examination, but it's quite a huge amount of data over the many years, and it's very hard to maintain such a big database, so in the new version that was released next week, we have done a compression, and we are just storing binary JSON fields with all the unique parameters of the e-mail, and we are now able to use the database about to one fifths of previous size. And, one advantage is that we had to make our template, e-mail templates versioned, so with every change in the plans there is now a new version created, instead of changing the old templates.

The small disadvantage of this is that this new database field is only available in a progress SQL, in versions greater than the 9.4, so this makes the FRED hard to use on very old systems, but the progress import packages for some old systems, so it's a little bit harder to install there, but it's doable.

One of the, let's say, [inaudible] things that we should have done earlier, but we did it in the last versions was that we actually

stored the EPP part source as a plate assimilated database and now it's hashed, even though these passwords are used as a second factor after the TLS [inaudible] certificate authentication, it's not a good practice definitely so it's better to have all the passwords in the system hashed. So, this is a feature of the new version, there is upgrades curved that provide all the changes so registrars, if you upgrade to new versions, it will not be attached in any way.

The interesting thing is that since a few weeks ago, all the source code is available on the GitHub, that we have always provided the source code to FRED as a source code package on the website, so we will not do that anymore; it's much easier to synchronize our internal repositories with the GitHub, so you can go to the GitHub and click there and see all the evolution of the code over time.

Some short things, more future plans, right now we have already started to work on the new web administration, this is something that we have not touched also for quite a long time, and we would like to update a little bit our price list to be more flexible to provide the possibilities to specify different prices for some different registrar groups and we know that we still have some defaults in the system that may reference our company in like photos or e-mails, something like that, so we would rather

like to have it a clean installation without our references to this, to our setup.

And, definitely there was always one of the features that FRED is missing that we haven't done for a long time, and that's a t-shirt, every software must have a t-shirt I think, even though FRED is one of the first software from CZ.NIC and this is the last one that has t-shirts, so I have a few of t-shirts here, so if you are interested in FRED, come to me and I will give you some. So, that's all for myself. I'm happy to answer questions.

PABLO RODRIGUEZ: Thank you very much. I actually don't want to take -- we're running behind, it's not your fault, but that's why I don't really want to take questions unless they are from the remote [inaudible], and I would like to move straight to Merike about IDN abuse. Thank you very much, Jaromir.

As I said, the name of the presenters will be in the agenda, and it's clickable, so if you want to contact just click and the mailto record will come up and go straight to the address. I have updated the agenda with the new e-mail addresses of Mark, so I will give it to Kim to post it.

MERIKE KAEO: Thank you very much, so this presentation is about IDN abuse. So, yeah, I will make it actually quite quick. So, I would present it, but I'm going to be presenting the work that was done by two of our senior developers at Far Side Security, so my shift men and Steven Watt.

So, the motivation for this work was that the engineers were looking at, you know, we have a lot of data to play with, and what is the abuse that is currently done using IDN's, or possible. And so, really, they were looking at domain abuse via IDN homoglyphs, and IDN's of course allow for forgeries to be nearly undetectable by the human eye, and isn't generally well understood by the general public in terms of what kind of abuse and hot much abuse is possible. We wanted to look at how widespread it is. Of course, IDN issues have been around and been talked about since the early 2000's, both in the IOTF and the ICANN communities, also in the security communities. But again, we wanted to understand what the prevalence and reach was.

So, I'm not going to go through these first few slides because I think this particular audience is very well versed with code points, Unicode, Punycode and all that.

I did want to just emphasize that I'll be talking about homoglyphs, which are basically looking at which characters

really look very similar. So again, I'm just going to click through in the time of expediency. Because, what I really wanted to emphasize is the research that we did, so we examined 125 top brand domain names, and they consisted of a variety of different folks; large content providers, luxury brands, social networking, cryptocurrency exchanges, etcetera. And, when monitoring these IDN homoglyphs in real time, we have real time passive DNS data, so cache missed data.

And what I'm going to show you is observations from three months of data that was sampled, and you can see the date is from October of 2017 to January 10th. So, there is a blog that was written, it came out a couple of weeks ago, and there was a pointer to that, so you will see the details, because most of the IDN homoglyphs that we found that were suspicious are listed there. I only list a few of them here in this particular presentation.

The thing that's really disturbing is that there is a large number of these homoglyphs seen; 116,000. And there is no assumption made at this point that there is a malicious intent against these domain or domain owners, that needs to be investigated. However, when the developers started to just randomly look at some of them, he did find a few live phishing sites; this is really disturbing. And what we did was we reached out, we

responsible disclosure to actually let people know, “Hey, are you aware that this is a potential live phishing site against you?”

And I will just state, when we first had the idea of writing a blog, I was very squirrely about it, right, because of the fact that are these maliciously used domains, these very significant brands, is this going to cause a problem? However, we did do responsible disclosure where we reached out in the security communities to every single one of them. Some of them weren’t listed in the blog, because they asked us not to. Every single one of them that’s listed in the blog was contacted, so they were fully aware.

So, for example, here’s Adobe and Apple. Right, again, these are suspicious IDN’s, I decided not to put the live phishing ones into the presentation because I don’t know whether or not it was still live or not or what was done with them, and so rather than cause more issues, but you can look at them in the blog and look at the references there.

So, these as you see are mostly from the .com GTLD but there’s also the .cf ccTLD. Now, Bank of America; right, there’s also quite a number of IDN homoglyphs that we’ve seen. Credit Swiss, and this gets really interesting, because you’ll see that these are also from a number of ccTLD’s, right? Then you have eBay, we’ve got Twitter, we have Walmart, and then Luxury Brand Social platforms, but the thing that’s really significant in

my mind is that you can see these IDN homoglyphs from any variety of TLD's, right? Either gTLDs or ccTLD's.

So, some general observations that I want to make here is that while IDN related abuse domains are a fraction of all the abuse domains, they exist. And, the publicity around this kind of abuse is growing. I even saw a blog from Brian Krebs, and once it gets into the mainstream, I'm always concerned about the fact that, well, how much is a criminal community now going to see how easy it is to use homoglyph, IDN homoglyph domains to actually now create abusive behavior?

And so, as I look at what is the role of the IETF, who creates standards and decides what characters can be used in IDN versus the role of ICANN, versus the role of registries, right? Who is actually responsible for what, and who is authoritative for what, and really, where is the policy enforcement needed? Because one of the things that really came to like for me, is while registries would set their own policy, are they enforcing it? And then of course within the ICANN community you look at, there are contractual obligations but not every TLD is contracted via ICANN. So, who is responsible, and what as a collective community are we going to do about this?

And that's really the point of my presentation; to raise the awareness of it and start further discussions.

EBERHARD LISSE: Thank you very much, we can take one question?

UNKNOWN SPEAKER: Thank you very much for this issue, if I go to your last, this is mostly about mixing of the scripts and the domain names, and we disused that in the past few days, which is except .com, nobody is allowed to mix scripts while registering a domain name. So maybe it happened in the past, but however I think somebody from ICANN or digital will answer that; is mixing up scripts still allowed?

Because these are all mixing scripts cases, it's just basically another IDN parts issue, versus when you are a string domain name, you are mixing the scripts and it is obviously creating an issue. So, if we have had this, probably we discussed it in the past, if this has then we have pretty much at this point. Of course, the similar issue will still demand there is 0 and [inaudible], all this will still remain there but mixing of script goes away then these many domains will have issues. Thank you.

MARIKE KAEO: Thank you for the comment.

EBERHARD LISSE: My comment on this is, whether it's allowed or not, if money can be made, somebody will try to use it. Anyway, okay, Don Hollander will be the last and then I'm looking for Jacques Latour, I can't see him -- there he is. When Don is done, Jacques will do the final presentation.

DON HOLLANDER: Thanks very much. Don Hollander from Brookhaven in New Zealand. When you looked at the potential attack on potential domain names, you only looked at ascii Latin characters, or did you look at potential homonym attacks in other scripts; Hindi, or Chinese?

MARIKE KAEO: No, right now it's just Latin characters.

DON HOLLANDER: And do you know anybody who is doing that work in other scripts?

MARIKE KAEO: I do not offhand, no.

DON HOLLANDER: Thank you.

EBERHARD LISSE: Okay, thank you very much. Give her a hand please?

JACQUES LATOUR: Good afternoon, my name is Jacques Latour, I'm with .ca. So, today I'm going to talk about IOT security frameworks. And, hopefully this is a call for help, so…I don't hear you? What line? Oh, at 9. DNS, these aren't firewall.

Alright, so why am I doing this and why do we need an IOT security framework? Because the number 1 risk for .ca as a TLD, many of you, TLD's the number one risk is a DDoS attack, like a large scare DDoS attack originating from IOT device. There's a real—today it's real that IOT device can be weaponized to do bad stuff, we've seen it recently with Memcached D.

It could have been worse, but the challenge is that once we add millions and billions of IOT device on the Internet, in an unstructured way, and those unstructured IOT devices that are typically unsecured are being used at scale to cause attacks against your TLD, then that can have significant impact on our security stability, so that's why I'm doing this, that's why I'm

trying to figure out how in the future, what do we need to do today to protect our future against large-scale DDoS attack.

Today is it happening; so, botnets are scouting for IOT devices to create better, bigger weaponization surface to create a bitter attack and reflection, footprint that causes big damage. So today, what's different today than different in the past is that I think the IOT device, people don't see them.

So, if you have let's say in the future 1000 water sensors in your house, and each one of them is being used to attack people, you won't notice that. Like, people don't notice that your water sensors are being used to attack you. If your computer is used, is slowing down, people notice your computer, your iPad, and stuff you use on a daily basis. But the scale of IOT device we're going to have eventually is the threat that we need to address, and we need to put the framework around that.

And, the framework is, so what I'm pushing here is two different ideas for the framework. One is; that at the bottom, the idea number 2, the secure gateway idea is for some reason, humans like to have everything connecting to everything. So, on your home network today, if you put as many IOT devices, every IOT device has full, unblocked access to the whole Internet; that's the way it works.

So, we protect the Internet from attacking your home network, but today we allow everything inside the home network to be fully accessible on the internet. That's something that we should do; we need to put some controls in there. So, that's the first part of the idea is that we need a next generation home gateway in the future that protects IOT devices from the Internet and protect the Internet from IOT device. That's something new we need to do at the home gateway.

So, if we're going to play the home gateway, the top idea is why not make ccTLD relevant in-home security for IOT? So, if we obviously, .ca, want to sell more domains or make them relevant forever, and then if we have the domain, the ccTLD is relevant in protecting home networks in the long term, then that's good for us for business as a TLD or ccTLD operator.

So, these two ideas together is what I wanted to message today. So, I think most of it is just common sense; we just need to implement better security controls in different things and make sure that in the future we're not attacked or impacted by the future threat of [inaudible].

So, this is the home gateway, I'm not sure, this is why I need help; I don't know who to tell or what to do or how do we make this a standard, but the home gateway today needs to be, in my view, needs to be better structured. There needs to be different

zones inside your home network, just like an enterprise network is zoned with different assets. And then you need to have accessed control to protect your internal and external assets.

But rule number one; which is, this is the one that's driving me the most crazy, is the concept that people can connect IOT device directly on the Internet. So, by default, an IOT device is typically something that doesn't have the means or the processing power to protect itself. It's a water sensor, or it's a camera, for camera for example they have great bandwidth that can cause damage. Do not connect an IOT device on the Internet directly; we need people, At Large, the entire planet needs to know not to do that.

So, you don't put mustard on lasagna, you don't connect an IOT device on the Internet; it's got to be that simple. People have got to know; I'm not going to plug it in on the Internet, so I'm going to make a t-shirt; no mustard unless I on lasagna or something like that, everybody knows that, that's super obvious, you don't do that. Anybody does that? No, okay good. Don't connect an IOT device directly on the internet always behind a gateway, a firewall, something that will protect it. That's my number one thing, if I leave tonight and somehow, we make this a standard, then we are good.

The other one is; segmentation. So, enterprise networks, serial internal network, you put your database in the zone, you put your application in one zone, web proxy in a separate one, you have access lists in between to make sure that if something is compromised you don't compromise the whole thing. That's standard enterprise security. Home networks should have the same thing in the future; you should have zones.

You put sensors in one, you put appliance in separate ones, you put cameras in a different one, and these are based on rules. If you are outside of your home network, you shouldn't need to see your home water temperature sensor; this device should only talk to other things inside your home network, and that's the kind of access that we need to have inside your home network.

And then it's just basic sense; sensors, that zone where you have water sensors and all that, it should not have access to the internet. Like, your water sensor shouldn't be able to do a scan of your Internet, every single IP address, V4, V6. You need security controls to protect the Internet from your IOT device and the other way around. So, most home gateways, they block traffic coming in. We need to control what comes out and what can talk inside your house. So, this is what, I don't know who to

ask; the ITF, or which vendor we need to talk to, but this is the common sense that all home gateways should be working on.

Because the challenge is, if we have a house, and we have a thousand cameras inside, and you have a 1 gig home connection, you can potentially generate a gig of traffic if it's not done properly and use that as an attack vector. So, if you have a million houses with 1 gig each, we're talking a lot of traffic globally distributed. This is what we need to address, this can impact any Internet operator.

So, we need to start thinking about controlling this access, so that's why I said I need help with this; we need to figure out a way of making our gateways smarter and more secure.

So that's it for the gateway part. This is the idea around the home registry. So, if we're going to have a new home gateway, with new access control and all that, so the idea came, "Well, how do I provision a fridge in the proper zone," like what would be the standard to provision a different IOT device inside your home network whether you want a light bulbs to be a certain attribute, you want your fridge appliance to have a different type of security control, you want your TV, your multimedia, to be reachable from the Internet when you're outside, but only for you to VPN and all that, so I said, maybe this is an opportunity to

do some brainstorming and come out with the framework on how we might make this work.

So, DNSSEC, take a little bit of DNSSEC, some innovation, some DNSSEC keys, we can link the home gateway with a domain at home, DNSSEC at home, and then the ccTLD, we provision a domain for the home gateway, we sign it with DNSSEC, internally you have a domain name structure that is unique to your house. You can access your home network remotely through the internet and then with DNSSEC it makes all the magic and it all works securely, so that's the dream.

So, I'll cover that in more details, but the goal is; as ccTLD, we want our future to be bulletproof. So, if the future of securing IOT is with a domain per household, then that's a good model for us to work in. We can support that around the planet if all the ccTLD we do this the same way then we can create a better future in front of us, taking some uncertainty out and make it more structured.

So, if you look at this picture, your home, a lot of your home attributes need to be accessible inside and outside of your home network. When your car is going to drive itself, the inside of your car is going to be part of your house. And how does the inside of your car connect to the inside of your home to see what's inside your fridge? So, you'll need to be able to VPN or remotely

connect in a trusted manner a lot of different zones together and make it work. So, the idea here is to create a framework to support this kind of communication.

So, this is registry automation, and home network automation. And, a lot of innovation around this, so we're slowly building the building blocks for this at CERA [ph.] to build a prototype, but I'd love to have a prototype of this running like next year.

So, this is a story; so, the last ICANN meeting, in Abu Dhabi we covered the high level story here, but here I'm refining it a little bit, so I'll go through a couple of slides on how I see this working, and how I see…and then obviously I'd love to have some Internet drafts and new protocol and new functionality out of this. But, this is the story; step 1: so, you go to Best Buy or the local computer store, and you buy your home gateway, and it comes with a domain bundled.

So, for .ca, it comes with a .ca domain. It's bundled with the domain, so you buy that, then second step: you just take that box, you open it, you install the CERA Home Gateway App, so you install the app on your gateway and then you turn the home gateway on, and then you tap your mobile on your home gateway. So, you tap, and then the app says, "Hey, what domain do you want for your home gateway?" So, you put a domain name, you search, it's available; myhome.ca, cool. It says, "Type

your secret—" so, there's a card that came in the box, you tap in on your home gateway, and boom; that domain is provisioned on your home gateway.

So, the home gateway then talks back to our provisioning engine, so in the registry we create a domain, we know how to do that, we sign it with .ca, with DNSSEC, we're primary for that domain on the internet, that means it can be resolved, it works on external. And then, it's fully operational within the registry and the internet. Then, it connects, the home gateway connects back to the registry, and we provision the domain inside that gateway with sender keys inside the home gateway securely so that same key for that domain is in the home gateway and the registry for that domain. And then, it's provisioned; that means with dynamic DNSSEC or dynamic DNS or something like that, your home gateway is connected to the registry with your home domain.

So, the important thing here is having a chain of trust in your home gateway so that you have a secure environment to do a lot of cool things I think.

So, that's the platform; there's nothing here we can't do, it's all—there's no rocket science, we can do all this today. But, your home gateway is connected to the registry, and then DNS is in sync external and internal.

So, at this point, you don't have much. So, now you need to set up your network, so the concept of having somebody type a Wi-Fi password, put in your own Wi-Fi password, that's gone. So, the first thing you do is you tap your application on your home gateway, and then your mobile learns the Wi-Fi network; nobody types W-iFi passwords every again. You tap and trust and move it, it's all automatic, nobody types it.

So, your home with your mobile app that's trusted to manage your home gateway. You click your phone on your mobile gateway, and your phone is now Wi-Fi enabled. And then you walk around the house and then you tap your trusted device. You tap your TV, your fridge, your car, and they all get the Wi-Fi password to connect to your—maybe not the car, but all the devices connect to your Wi-Fi network; that's how you transfer the keys around, through the app, through your phone. Nobody types passwords anymore for Wi-Fi. The Wi-Fi password changes automatically, you don't need to take care of that.

This is, not the future, very distant, but this is something we can do also, there is no rocket science here.

So, at this point, you have a home gateway, so we've got the domain name provision in your gateway, it's fully signed with DNSSEC internal and external, same key, and now we're at a point where you can provision devices. So, externally on the

Internet you can add VPN.myhouse with your external IP, that works. Internally you can add devices.

So, the idea here is that once you grab your phone, your app, and you can discover an IOT device and the IOT device exposes the service that it has. So, with your phone if you tap with the app, if you tap on the fridge, it's going to say, "You know what? This fridge has a status, a camera inside your fridge, and it's got alerts." And then you drag which services that you want inside your gateway, so your gateway is actually monitoring your fridge, it's proxying the video camera or something like that, and the gateway is in control of all your IOT device inside your home.

And then you configure on your device when you discover your IOT device, you tell it what services you want from that device to integrate in your home networks. So, your fridge is already on the Wi-Fi network, here you pick, "I want to see the status of the fridge," and then in your IOT dashboard you get the status of the fridge, the door and different things inside your house.

So, the idea is that you drag and drop devices and services by type, and this gets provision in the back with the proper zone. The security zone where that IOT device should stay, so your fridge in this case would go in an appliance zone. So, for example, if you want to grant somebody access to your network

inside your network, so you can grab a user, and say, you drag the VPN, the remote access functionality to that user.

So, that user by default can VPN inside your network, you don't have to give it keys, there's no IP sec, you just drag and drop services by tapping and dragging services. So, the idea here is that there's no manual configuration and everything is automatic, you drag and drop, but then you create, when you assign functionality by device, that's where the zoning, that's where the zoning security controls are implemented.

So, this is the last example, so if you click on the car, the car you can control it, the doors maybe, the alert, the status, then you can drag remote access on your car, that would grant your car remote access to your internal network, plus different services, so the idea is that again you don't type or configure manually, you just drag and drop and set up your home gateway to have the right security controls to make sure that everything is typed and not like today wide open and full of risk.

I can read that later. So, we're building, we've got a few on home gateways, we're going to start playing with this, building the app, exposing some services through our FID for like fridges and stuff. It's all, everything I have is public on GitHub, it's here. So, building the IOT network, that's the vision of drag and drop, and create a secure framework. The first part, building a secure

home gateway, this is something we need to do. This is for our interest, we've got to make sure that we cannot get—we need to mitigate the risk of DDoS attack on TLD operators, this is one thing we need to do together to make sure we have a brighter future. So, that's it.

EBERHARD LISSE: Okie, doke. Thank you very much. I will allow one question?

HOWARD BEN: Thank you, Howard Ben from Samsung Electronics again. The IOT industry I think is very, very aware of these topics, and in fact there are two standard organizations that I think you should be looking at; so the first one is OCF: the Open Connectivity Foundation, they are really focusing on the devices themselves and how they interact with the hubs, and then there's one: M2M: both of them have got websites I can point you to, I think openconnectivity.org and 1M2M.org are the two web addresses, they've both got open source projects associated with them. They solve an awful lot of the problems that you've explained here. These are, as I say, very, very well-known issues, so hopefully that will point you in the right direction, and we can talk afterwards.

JACQUES LATOUR: Okay, yeah, will do.

EBERHARD LISSE: Okie doke, thank you very much. That leads me to hijack Christian Hasselman who without much notice has been Shanghaied to give the closure. And some thanks to Jacques.

CHRISTIAN HASSELMAN: Thanks Eberhard, so we have the tradition if you will to close the Tech day with a few closing remarks, and usually that's being done by somebody in the Tech Working Group, so today that's going to be me.

So, I think that the things that we saw today have, let's say, there's a strong emphasis in the presentations on security and resilience. So, for example we saw that from different perspectives, I think we saw it from an end user perspective in the presentations by Marike on homographs and IDN's and we saw the presentation by Patrick this morning on emojis in domain names, so these are from an end user perspective.

We saw quite a few presentations on security aspects of the DNS system itself, so we saw a presentation on the local route, on the DNS defense layer, and also on KSK sentinel by Warren Kumari, and we saw presentations on internet resilience in general, so

not specifically tied to the DNS but on let's say the availability of Internet connectivity.

So, obviously the presentation by our host, .pr on the devastating effects of the hurricane on the telecommunications infrastructure. We saw the [inaudible] presentation by Google to restore that or help restore connectivity here in Puerto Rico. And finally, we saw the presentation by Jacques, at least the first part of it, that was aiming at avoiding DDoS attacks and thus keeping the Internet available.

So, I think this goes to show that the internet is a critical infrastructure as we all know, and I think that there might be a trend going forward that we'll see more security and stability related presentations at Tech Day, perhaps at more emerging topics like the IOT. And so, I'd like to call on all of you to submit your presentations for the next Tech Day at ICANN 62. And those were my observations. Back to you.

EBERHARD LISSE: Thank you very much. So that concludes it, and have a nice ICANN meeting. I'll see you then during the week, and in three months.

[END OF TRANSCRIPTION]