ABU DHABI – Tech Day - part 3
Monday, October 30, 2017 – 15:15 to 16:45 GST
ICANN60 | Abu Dhabi, United Arab Emirates

UNIDENTIFIED MALE: It's from abuse. We have currently 23 accredited registrars, both local and international. Currently, we have 210,000 registered domains, and as I said, we are the first adopters and we have a well-established model. We have easy registration, we have credit registrars and online registration, very quick, no paperwork needed online registration.

This chart just talks about the progress over six years regarding the number of registrations. We started with around 84,000, until today we reached around 203,000 in 2006 with 210,000 in 2017.

As an operator, we operate the DNS .ae and our registry system. We have a redundant infrastructure, we have a DR site, we are certified ISO27001 and business continuity ISO22301. As of nameserver, we maintain high availability. We have secondaries, we have partnership with providers such as PCH and IEC. We are planning to deploy DNSSEC in 2018. We are in study phase now, and we're planning to have it implemented in 2018 inshallah.

As of DNS performance, we're monitoring using monitoring system. We are currently using one of them is DNSMON from RIPE, provides historical data, up-to-date data how the DNS performance, how is the response time all over the world. It’s a very nice tool. Ourselves, we have one probe in our networks. This slide give us just a sample report of October 2017. [inaudible] everything is green, and this is a relative response time. And we have then another slide of showing the unanswered queries, which is very promising, very nice. I think this is the last of my slides. If you have any questions for me.

EBERHARD LISSE: Thank you very much. It was short and concise. I liked the ISO standardization. One can see that even though it’s not one of the largest registries if you have got funds available to design a proper operation and you don’t have to make do with what you have, but you can do it properly and implement standards which are expensive to set up for small ccTLDs like us with our 4500 names. Thank you very much. There is a question from the floor.

UNIDENTIFIED MALE: [inaudible] I will start my question in Arabic first, and then I'll continue in English. [inaudible]

I was asking the gentleman what are the main challenges in terms of administration and technical, and how are they overcome, because as we are a new ccTLD operator, we’d like to benefit from their experience. Thank you.

EBERHARD LISSE: What ccTLD are you operating?

UNIDENTIFIED MALE: .kw, Kuwait.

EBERHARD LISSE: Kuwait, okay.

UNIDENTIFIED MALE: As a start, because back in 2008 we took the operation of the registry, we had to do a lot of awareness, and we had to do a lot for the public, because it was with one registry and now you have a choice of many. So one of the challenges is people just kept not understanding what changed, how a legacy domain will be affected, what can I do, what's the benefit of having this new model? And we had attended a lot of functions, we worked at workshops. It’s mainly awareness, this is the challenge we had first.

UNIDENTIFIED MALE: Okay. We are –

UNIDENTIFIED MALE: [inaudible] same model registry, accredited registrant and the registrant at the end? Have you faced any difficulties from accredited registrant, like uploading or submitting documents, or correct information of the registrant?

UNIDENTIFIED MALE: Yes. We had before we had accreditation procedure, actually, before someone comes accredit with us, which is .ae. So we have before becoming accredited .ae, you have to go through many tests before being accredited. So that helps us a lot of the registrar being aware of what to expect from us and from them, like uploaded data has to be correct and what sort of information needs to be uploaded.

We faced some challenge at first with the help of registrars, they start to understand what – for example, sometimes they put some information that’s not complete or miss, so we just keep contact with them and tell them that please correct, and we didn't have – actually, we look at it as a partnership with registrars. Mistakes happen, so you are in partnership with an entity that benefits you, so in there's a communication. That’s what I see. As much communication we have with the accredited registrar, it will be a benefit for both sides.

UNIDENTIFIED MALE: [inaudible] Thank you.

EBERHARD LISSE: My own experience with moving to a registrar-registrant model is that too much information for the end user, for the applicants, is also not a good thing. The more they know about, the more they hear about it, the less they know about it is my view. So some information is good, but one mustn't give them too much information because they don't understand what we're doing anyway.

So as long as it works, and we had a reseller model with one single registry and we pushed them off and lowered the prices by going to other registrars and kept the prices over for the old one, they would eventually figure this out. But my advice would be to inform the end users, but mainly if you move to a registrar, it's going to be cheaper for you. If you don't, it's going to be more expensive. Most people will understand that.

To tell them registrar, registry, they don't understand it. They don't care. They want to know Facebook, basically, and Twitter, and their own website for the company. And the e-mail. That's all they want to understand as an end client. So tell them, "We changed our model, but it'll only be cheaper for you." And if they want more, then they'll send you e-mail and you can explain it to them in more detail. But those are the ones that will actually understand and are interested. Most clients don't change their ISPs no matter how bad they are, no matter what the prices. We see this even in our country which is developing. They don't care, they don't change their service provider as long as it works, no matter whether it works good or bad or whether they pay more. It's not really the main interest, they don't want to bother with it.

So my point to this is this sounds like a very well thought out operation and way of doing it. That's something that you can learn from, especially if you have the resources to implement the standards, but don't make it dependent on the understanding of your clients. They will figure it out. Thank you very much.

Now Jacques will tell us about the Internet of Things in his house.

JACQUES LATOUR: Good afternoon. My name is Jacques Latour, I'm the CTO at CIRA for .ca. Basically, today I'm going to talk about an idea, that's all it is right now, and I want to share it with you guys. If you think you like it, then I'm looking for help to build it and make it happen. So it's not a typical tech day presentation, but at the end you should see there's some value I think for you guys, for all of us.

So it’s the home registry ID. We’ll talk about that. Basically, if you look at the way the IoT network are implemented today, the security around it, I don’t see a lot of framework. So the implementation is vendor specific is disparate, there's no structure. What I'm proposing is something that is structured around the security of IoT for home networks.

A home network should be safe. So in the future, think about in five years from now or in ten years from now when you have an IoT infrastructure, everything is connected, that’s what I'm talking about, the future. We've got some time to get there, but in the future when you set up an IoT network, it should be simple, so my grandmother, my mother should be able to do this super easily.

And the home network needs to be reachable from the Internet. That means whatever your house is not your home network, so it needs to be remotely, securely accessible from anywhere on the Internet. So that’s one key criteria. Your car should be connected to your home network, so your house is not your home. Your home on the Internet, much bigger, so everything that in your car when it’s self-driving, you should be able to click on the dash and see how many eggs you have left, and that should be automatic. That’s what IoT, it’s everything connecting to everything inside your house. That should be secure. So your home is bigger than your house, it’s your Internet home we’re talking about.

Now it’s going to grow and then you have personal device, so you have personal, wearable IoT device that needs to be part of your house. So if you have a camera on you and then your wife wants to see where you're at, technically she can look at your camera and see what you're seeing, or whatever. So your house network, your home network now includes stuff that’s connected to you.

But the key thing is in the future, you want these devices, your home network to be usable, visible, accessible only by you. You don’t want anybody else to access your stuff, and the challenge is that your stuff is all over the map. It’s in your car, it’s on you, it’s in your suitcase, it’s your belt that’s telling you you ate too much. All of that stuff inside and outside, it all needs to be one trust network on the Internet, irrelevant of where you're at.

So very easily, the solution is so obvious: you need a common key to make everything talk together. That’s how you create the trust. So your home network has a single key that everything you own, that you trust, is on that trust network, that realm of security that you enable. So that’s the vision. Long-term, it should be super easy. Anybody from five-year-old to 99 should be able – by then will live to 120 – to set up this kind of infrastructure without being an engineer.

So obviously, it has to be IPv6. Why is this important to us? I think it's super important. The reason I'm doing this is that for the domain industry, for ccTLD, I want to have a domain name per household. In the future, we want ccTLD to be relevant in the future of IoT, and you want a domain name per household. So if we do that and then we use that framework to secure IoT in the future, that means ccTLD remain relevant in the future of the IoT.

So if some companies, Cisco, Belkin or whatever decide that the future of the security of the Internet is using .whatever and everything switches to that, then we'd become less relevant. Eventually, something is going to need to happen. So if you want ccTLD to remain relevant in the future, then you have a domain per household, and then we build all the magic and the glue to make it work around that. Then we're in control of our destiny, and that is obviously an objective here.

No questions. I've got another 40 minutes to talk [inaudible]

UNIDENTIFIED MALE: So since DKG is not here, I will pretend to be DKG. You're putting all the information that is really personal in a publicly queryable database.

JACQUES LATOUR: You don’t even know what I'm doing yet.

UNIDENTIFIED MALE: I agree we don’t know what you're doing, except that you're using public DNS, which is the one thing you shouldn’t be using for this.

JACQUES LATOUR: I agree. I'm not doing that, so keep watch. We want to be relevant. How many of you agree you want to be relevant in the future forever for IoT?

EBERHARD LISSE: Since you're both living in Canada, why don’t you go visit his house and have a look?

JACQUES LATOUR: So that’s the vision. This is all based on DNSSEC, some chain of trust, some innovation, and then this is the key to create home platform. So if we want to be in control of our destiny, we’re ccTLD, this is the path forward, this is a vision to go. So no more questions.

So the thing that drove this a little bit is the IETF is working on the home net, so they try to figure how to delegate – what domains should be used in households by default when you turn your router on by default in its home.arpa. And you can't turn on DNSSEC on that because you can't create a chain of trust if everybody uses home.arpa at home. And technically, if you use this large scale, you can't use DNSSEC. Any innovation in DNSSEC at the home to make your infrastructure more secure.

However, all that work for home.arpa, so the IETF is focusing on, “What if you don’t have a domain? Then use this default.” I'm saying we should focus on, “If you have a delegated domain that can be signed with DNSSEC, then let’s figure out how to do all of this.” So the solution is, let’s use delegated domain, .ca ccTLD domain for your house as a solution. So it’s automation. So for this to work, it’s registry automation, it’s home network automation, and lots of innovation. So we take all this stuff together, all the knowhow we have. I'm sure we can build this solution and make it work.

So the solution is your local ccTLD will provision your domain, sign it with DNSSEC, establish a secure chain of trust to your local home router, magically – see, I have a solution – solve all your worries and keep your family online safe. See? It’s all there. It works. It already works.

So that’s a napkin idea I had many years ago, that’s what a diagram I made in Visio using crayon art. This is how I wanted it to work, so that’s an idea.

The next slide, I want to run through how I see this working, and then in the end we can talk about how we really make it work. So I'm a high level architect guy, I'm not a super detailed, hands-on guy, but this should work. So this is a story of how this works.

Number one, you go to your BestBuy. I don't know if you have that everywhere, like electronic store. So this is a story for .ca, how I see this happening. So you go, you buy your .ca home gateway. It comes bundled with a .ca domain by default. So you have a box with a home router and RFID card thing with your domain. So you didn't think about that so far, right? Next.

Alright, so now this is a step-by-step procedure to how I see it to provision your home gateway. So you open the box – oh, first of all you install the CIRA Home Gateway app, the little app. So you go to cire.ca, download the app. Then you turn your home gateway on, then you tap your mobile on the home gateway so your mobile is trusted to manage your home gateway. The last thing you want to do is someone to log in a web interface and set up a home router and all that stuff. You tap it, you're done.

Then you pick a domain name that you want for your home. Perfect. and then you enter your secret code, so you tap your RFID card on the home gateway. That’s all you do. Boom. And then it’s ready for configuration.

So we have la-house-a-latour.ca, that’s my home network, and that’s what's going to be provisioned on the gateway. So if you can tap stuff, then you can tap it. I can see Rick thinking stuff there, so that’s good. Perfect. So now you’ve got the home gateway [to ready.] So what happens after this?

Can you just scroll one line down? Okay, perfect. I hate that, when you lose the... Let me go back. Why is the screen always… Okay, so I'm clicking. That’s annoying. [inaudible]

Okay, so now you’ve got the box set up. Then the box is going to connect to CIRA. The box is going to tell CIRA through that API that this domain, la-house-a-latour needs to be created in the registry. So we create the domain name in the registry, we sign that domain with DNSSEC, and then we’re primary for that domain, so we’re the registry for it, now we’re the primary and external DNS view for this domain, so that’s external service delivery, and then you do secondary Anycast services for that domain.

For the registry, you register the domain with us and we’re a primary operator for the external view of that domain, and we run la-house-a-latour.ca and we're responsible for the external use of that domain. So the domain is provisioned. Nobody did anything so far, it's all automated. Boom. And then the registry goes back to the gateway and now you need to set it up. So you establish connection back to the home gateway, you send the private keys up there, you set up the internal DNS – so the network internal DNS is operated by the gateway. The external DNS is operated by the registry.

So if you want to connect your TV at home, then you go tv.la-house-a-latour.ca, and that's resolvable on the Internet to go to your home gateway IP so you can connect it. So there's some sort of dynamic DNS integration, synchronization between the external DNS and the home gateway DNS.

So everything we've seen so far is all doable, there's no rocket science at all. It's just linking all the pieces together and make it work. But so far the end user, all he did is he tapped twice. Tapped his phone, tapped the card, put the domain name, click enter, boom. It's live. So now that the home gateway is all set up, secure with the DNS key, that's where we can do innovation. There's lots of stuff we can do from there.

So now you need to set up your network. The idea is that you use the app and you say, "You know what? Put the box in green mode," and you tap your phone on the app, and the app programs the Wi-Fi. So you don't even know what the Wi-Fi password is, you click your app and your phone is connected to the Wi-Fi network. So the keys are passed automatically. You don't need to write down a Wi-Fi key, it's all automatic.

Then you grab your phone and you tap the TV, and your TV, you give it the Wi-Fi code. Now it's part of your network. You tap your fridge, your car, your belt, your glasses, whatever, all the IoT devices you want in your house, you add them by connecting everything and creating a trust.

That's in my view in the future when you create a secure network, that's probably the only way you can add thousands of devices, is by adding them by tapping. You can't go on the fridge or on your belt and set up the Wi-Fi or whatever [inaudible] community you want to have. It all has to be simple to set up yet secure. So it's easy-peasy, I learned that's an English word to say super easy to do.

So remember, it's an idea – like Ikea, idea, I have to remember that – and it's a feasible thing. So it looks like this. The reason you want – the registry is in charge of provisioning this home network domain with the gateway, and the gateway is in charge of provisioning the internal DNS, managing all that and managing the security or infrastructure. But the key thing is that the gateway is always synchronized with the home network registry to make sure that if you want to make your TV available on the Internet, that the name you want external is published in the home network registry.

So an example of – so if you have your phone and you tap the gateway, by default you get the IPSEC keys. So if you're outside your house and you want to see how many eggs you have in the fridge, you click on your fridge and it sets up an automatic IP sector now, because it knows all the keys to VPN in, connect in and communicate. That's an example. And then if you buy a lamp, like a lightbulb of some sort and it absolutely needs to talk to GE to update its firmware, then you can allow – there should be an app to allow GE, like the cloud service to access the gateway.

Since we're CIRA and we have [D-zone] or firewall recursive DNS service, this gateway will have that clean DNS service where there's no [inaudible] and all that. And then you can even set up kids in your house that have super clean DNS service with restricted access to the Internet. So that's the idea of making a secure IoT platform in the future using a network trust all based on DNSSEC with full automation, and it keeps ccTLDs relevant in the future. And in my view, that's a possible thing that we can all build, so there's no rocket science in there, this is something we can actually all build and make work and write a bunch of IETF drafts and standard to make this the de facto standard to secure IOT network [in the home] in the future based on ccTLDs.

So the delivery model can be use your own routers, or you have a deal with Cisco and vendors or whatever, that's to be defined, but the idea is to have at least a vision of where we need to go. It looks approximately like that. So we'll have some questions.

Like I said, it's a journey, so we want ccTLD relevance in the future, we want to keep networks safe, your Internet home is bigger than your house. That's it, right? So the last slide is we're going to start building a prototype with the Czech Omnia router and go from there, and make everything public. That's it. Questions?

EBERHARD LISSE: Thank you very much. I like this presentation. I don't like one word on the last page. "Sell CIRA home gates." I'm all for open source and giving it away for free, but that's a separate issue. You're going to be more than happy when your fridge is going to a tech – [done] again that you actually make money off it. Is this not something where we could write an RFC on?

JACQUES LATOUR: What?

EBERHARD LISSE: Is this not something where we shouldn't research but write an RFC on?

JACQUES LATOUR: Oh, yes, we need to write lots [inaudible]

EBERHARD LISSE: I would be really willing to assist you in crosschecking and wordsmithing it. From the floor.

BARRY LEIBA: Hi, this is Barry Leiba. You might want to discuss this if you haven't already at the Thing-to-Thing Research Group. One thing that we discussed in one of the Thing-to-Thing Research Group meetings was when you sell your smart house, what happens? How do you disconnect things? Your smartphone, your car and your Fitbit, whatever, go with you. Your lightbulbs and maybe your refrigerator, and maybe your washer and dryer stay there. Maybe some of them don't, maybe you take some with you. So untangling it all, disconnecting it and making both you and the new owner happy that it's all secure is an important thing.

In that discussion, one thing that came up is that things overlap. So your model seems to have – all these things are part of your extended house. But maybe your car is also part of your parents' house, but only somewhat. That kind of thing. Your smartphone gets you into your house without a key, your smartphone may get you into your parents' house without a key, but doesn't get informed every time your parents' refrigerator is low on milk. So it's a partial connection, and I'm not sure that fits into this. I think this is the core of a good idea though.

EBERHARD LISSE: Let me just quickly – among other things, I'm a German citizen. Germans only buy houses, they never sell houses.

JACQUES LATOUR: So the answer to that is if having a device belonging to two different networks is the issue you're seeing, then we're on a good track, because that's something we can easily deal with. Thank you.

BARRY LEIBA: I guess the extended thing of what I'm saying is that this is a good start, but there's a lot of thinking to do about how it fits together easily with tap, tap. That's all.

JACQUES LATOUR: It's a journey. Absolutely.

EBERHARD LISSE: And the older your parents are, the more important it is for you to know that their fridge is getting empty.

RICK LAMB: Rick Lamb, not representing ICANN. Great stuff, and maybe the counterpoint from the previous speaker here: just do it. You'll figure it all out.

JACQUES LATOUR: Oh, yes.

RICK LAMB: Just start the process. I've done things like this before using DNSSEC as the mechanism for securing things, and as you know, there are others out there that have used the – what I see here – and tell me if I get this wrong – is that you'd extended the DNS model in the hierarchy and, say, the DNS server essentially lives in that little gateway. Correct?

JACQUES LATOUR: For the home user [inaudible]

RICK LAMB: For the home user. So you're extending the naming scheme there for other things. And I agree a little bit with Paul that maybe you don't want to have too much information there. That naming scheme could be not necessarily rickskitchen.something, it could be some opaque string. I know that makes it a little stranger, but it could be an opaque spring. If I didn't want people to know that I had two refrigerators for example, or whatever.

But other than that, this is really great stuff. I think a lot of the overthinking this and trying to get all the designing the standard first or going through that first I think is a mistake. I think it's better to learn as you go through the process of developing this particular idea, product, what have you, and I also like the idea that it's a business opportunity for the registry in this case, because it's way too complicated for grandma to deal with. And if you could somehow act as that interface and make it all simple, tap, whatever you want to use, I think this is a great idea.

I think starting with the .cz Omni is a wonderful idea, but it's also very easy to build this sort of thing into some of the existing devices. And that's something I guess I would like to talk to you about that you should know about. A lot of these things like the lightbulbs, the on-off switches are all based on the same set of chipsets, they're actually very easy to reprogram and reburn. So if you wanted to do a demo of this, say at the next ICANN meeting or something, you could just show up and have all the hardware build already and say, “Look, here's an example that I'm just making this all work.” Wonderful stuff. Thank you.

JACQUES LATOUR: For the next meeting? No?

EBERHARD LISSE: One more from the floor.

JOHANNES LOXEN: Yes. This is Johannes Loxen from Sernet, a German-based security company. I don’t trust smart devices in my home until now, so I just play around with that, and this idea reminds me a little bit of [Enom.] Nice idea, but it didn't work because competing companies want you to force into their data silos. I have on this iPhone five apps that let me control, in principle, smart devices in my home. One of that I use is for my Fritz!Box, my router. This is all I trust and which I can control. The others are things like a heater that wants me to photograph a QR code and then by some magic lets me control my heater. Or my Volvo car app forces me to go to volvo.com and then magically control my car. So I have no chance to identify any IP in order to put it into my DNS. I'm just in their data silos and I have to use an app that lets me directly to my machines, so no way to get it controlled in my home. So I will never use my Volvo car in that home scenario. This is a challenge.

JACQUES LATOUR: So you're talking today, this is future. Today’s implementation is really bad. If somebody [acts in] Volvo and they have access to all the cars, they can start and stop every car they want.

JOHANNES LOXEN: Yes, that’s right, and this is the interesting thing –

JACQUES LATOUR: So this is not [that.]

JOHANNES LOXEN: [inaudible] start my heater and stuff, and the thing is what the industry will decide to go. Yes.

UNIDENTIFIED MALE: I would just say you can just drive a BMW.

EBERHARD LISSE: Okay, any other questions? We still have a little bit of time.

JACQUES LATOUR: The slides are available on –

EBERHARD LISSE: Okay, we take these two questions from the floor and then we are done. Ladies first.

NARELLE CLARK: Well, yes, speaking of ladies, I work with domestic violence victims.

EBERHARD LISSE: Sorry, who are you?

NARELLE CLARK: Narelle Clark, ACCAN, the Australian Communications Consumer Action Network, and I'm working currently also with [inaudible] So in the Australian context, we work with some domestic violence victims. How can you implement protocols in there [as to] ensure that when a family – or when half the family, shall we say, or three quarters of the family – are feeling a violent situation, how can they then suddenly cleanse these systems of their data? How can they lock out the perpetrator of the violence? Have you thought about these sorts of considerations? And how will we work that through? Because this is a really serious tool for gaslighting and suchlike if you really want to go there.

JACQUES LATOUR: Yes, so there would be something on the gateway that would control content maybe. If I understand – because it's very hard to hear with all the echo here –

EBERHARD LISSE: No, it was a question about domestic violence. As you may know, I'm a gynecologist. I have some actual professional experience with this from a professional aspect, and this is an important question. If somebody in the future sitting in such a house wants to leave, what consequences, how do you turn this off, how do you do this? That's something that one would have to consider in the design phase. It's good that you point this out. This is a non-obvious thing for a techie, but as I said, I'm quite interested in this myself. So the more ideas that go into such a drafting process to come up with a design is probably helpful. On the other hand, one doesn't want to overdesign it.

JACQUES LATOUR: So that's all in the app controlling who has access to what and defining what your trust network is. But an example I have there, I've got a firefighter, and the idea there that when I did a presentation before is they should have some access under very special circumstances to access all the cameras in the house to see where the fire is or where the threat is. So that’s [an idea.] I'm not sure how you implement that, but the idea is that if your house is on fire and you have a cat, maybe they're able to look through the camera if you give them access to see where the cat is. So that’s an option in having a network that you control, that you can give other people access to temporarily. So if you give access to people, you can also remove access to other.

NARELLE CLARK: You need to revoke that access very quickly in the context of domestic violence, and you also need to protect the family of the firefighter who might be a perpetrator as well.

JACQUES LATOUR: [inaudible]

EBERHARD LISSE: And the last question.

RUSS MUNDY: Hi, Jacques.

JACQUES LATOUR: Yes.

RUSS MUNDY: This is a very interesting idea. I've looked at things of this nature before, laying out a DNSSEC approach to how you might solve a problem of this nature. Did a little bit of – I don't remember exactly how long ago it was, zero [conf] guys in the IETF for a point of sales effort. At one point, we laid out how you could design and put in place the steps necessary to build your initial set of trust for a point of sales device that was going to be manufactured somewhere and would end up being placed, and the extent of the knowledge of the installer was how to look at a picture and plug two wires into the back of a box, and be able to come up securely.

Similar sort of problem for achieving the initial secure state, but this is really much more complicated. I think Barry hit on a good point earlier. Things overlap, things have to change, so in some ways it almost seems that one needs to consider the possibility of having effectively a certificate authority type of function associated with this layout of all these pieces that belong to the virtual house.

JACQUES LATOUR: Oh, yes.

RUSS MUNDY: So DNSSEC can do a lot for that, but I'm not sure that it’s feasible for the entirety of the solution. You need more access control mechanisms that are better, fine grain control and so on. So I think it’s a cool idea, and keep pushing on it, but think about the secure leaving, joining, partial overlaps kind of things.

JACQUES LATOUR: Yes, for sure. Well, Paul is going to figure it out and then it’s going to work. So I'm not worried at all. That’s it, thank you.

EBERHARD LISSE: Okay, thank you very much.

JACQUES LATOUR: And we have a bunch of .ca hats, so if you want them, they're on tables. I don’t want to bring them back.

EBERHARD LISSE: Okay, now we hear about Arabic script IDN.

RAED ALFAYEZ: [inaudible] Hello, everyone. I am Raed Alfayez, this is my second presentation. This time, I will speak about supporting Arabic domain names. SaudiNIC has a very long journey for supporting

Arabic domain names, and I would like to share with you some of the things that we have found, and maybe you'll carry on our message to other people, other engineers, especially in the [ITH]I for other registries.

So this is my agenda, I will give some information about the SaudiNIC and then a small introduction about Arabic domain names and the Arabic script, and our efforts in supporting Arabic domain names and what is missing. So again, SaudiNIC is the registry administrating the domain names for .sa since 1995, and .saudi since 2010. We are operating under government organization CITC, Communication and Information Technology Commission. We are leading body for supporting Arabic domain names, and we have more than 15 years of experience on how to support Arabic domain names.

We have more than 50,000 domain names, and around 2300 Arabic domain names, and we are the biggest Arabic registry. It’s a very small number, but there are so many obstacles that stop us from expanding. And for the people who don’t know Arabic, Arabic language is ranked as the fifth language by native speakers in the world. There are almost 300 million users speaking Arabic language, and it is [inaudible] orchestrated official and coofficial language for 25 countries.

We have 32 main characters in Arabic language, and we have some variants within it, so some of them have three or more, and some of them have just only one variant. The second part is the Arabic script. Arabic language is part from the Arabic script. The Arabic script is the second most widely used alphabetic writing system in the world. It is used by many languages, and the Arabic language is one of them, and there are Urdu, Persian, Turkish, Pashto and many other, more than 53 languages using the Arabic script. It is used in 43 countries, and the potential is more than one billion potential users who can consider using Arabic script domain names.

So these are some of the characteristics of the Arabic language and Arabic script. The writing system is from right to left. It has combining marks and diacritics [inaudible] and we don't have – the space is the word separator, and we have another – some are using a zeroth joiner, a ninth joiner and a hyphen. We have three sets of digits in the Arabic script, so we have the European digits, the Arabic-Indic digits, and the East Arabic digits. And there are so many similar characters, very huge list of similar characters which make our life a little bit difficult.

So this is the [inaudible] of the Arabic script. The green one is the Arabic language. You can see it's very huge script, and this is just only one block. There are other two blocks like this one. There are so many similar characters, and we group them like Kaf group, Heh group, Yeh group, Alef group. Some of them have maybe five code points that are similar to each other. Sometimes, they're exactly identical, so a mirror, exactly the same character, different code points.

So this all together, we have a problem that if you write a domain name in Arabic, there will be so many variants for one label. Let's consider Google.com, if we just only consider the domain part, not the TLD. I can say that there are 64 variants of Google depending on the upper and lower case, so if you type in the browser Google with a capital G at the beginning or make the second O is the capital and the rest small, you'll reach the same site. All of them are hosted, all of them point to the same IP address. If you send e-mail to [inaudible] everything is done through the DNS. But this is not the case for Arabic.

You can see there are two words. They're exactly identical. The difference is the color of the first letter which is Kaf, and the last letter is Yeh. So exactly mirror, two identical strings, two different code points within the string, and if you type it, you might reach two different [inaudible] sites. So you need to protect it, it's not like the capital and small. And this is the problem. This is why we are here and why we are facing many problems in this area.

So what we have done, we have started long ago and we have many dimensions. We started with Arabic IDN pilot project, we have built lots of tools, algorithms and solutions to manage variants. We have done IDN assessment reports, and we have launched Arabic e-mail address project. I will give a brief about each one of them. The first one that before IDN was implemented and the Arabic domain names were introduced or we get the fast track from ICANN in 2010, but we were registering domain names before. We started in 2003, and we do it internally within SaudiNIC, within Saudi Arabia, and then we joined project with United Arab Emirates and we launched in 2004 the GCC pilot project for Arabic domain names, so all the GCC countries have Arabic domain names working within the GCC countries. And then we expand this project to have Arabic domain names working within the Arab League. At that time, we have alternative root servers, that's only if a user type Arabic domain names, it will be redirected to alternate root servers. Otherwise, it will go to the 13 regular root servers. And with e have language and variant tables. If you need more information, we have org site called Arabic-domains.org, it has the historic information, all technical documents, everything. We have participated in writing an RFC. It's linguistic guidelines for the use of Arabic language and Internet domain names, and this is the link for the RFC. So this is history. Also, we have published the IDN – the language table and variant table in the new IANA xml format, and it is published on both .sa and saudia TLD, and also within the IDN guidelines for the Arabic Script Working Group.

So the second part, we also participated in building tools. I will go roughly through them. The first tool we started, this tool is available publicly if someone want to know the shape, because the code points for the Arabic have different shapes based on their position. So if it is in the beginning of the word, have a shape, maybe in the middle have different shape, at the end, and isolated. So there are many shapes. So in order to make people understand the shapes of the Arabic characters, we built a tool, we call it a comparison tool for Arabic script letters. It shows you all the code points. Click on it, it will give you different shapes with different fonts, and it will draw you an image for it. This will help us to see what is the behavior for this character. Because believe me, one of the characters, it's called – like noon with three dots above. However, if it is in the middle, the dots above will go down. I don't know why, but maybe the Unicode guys know the answer. But for us, we don't know this character, and the behavior was not – we cannot predict it, this is the problem. So these tools help us to know what is the behavior of each code point.

We have built the master key algorithm. We believe that since I told you that we have so many variants in the Arabic, it's difficult to store all the variants, and I believe one of the registries in the new gTLD have a solution, and there was a limitation to the solution. [inaudible] came and want to register their name in Arabic, and they said, “Sorry, you cannot register this name because you have more than 200 variants.,” The problem is that I will show you this table, etisal is a word like calling. It has 300 variants. If you put etisalat like calls or communications, it'll be 6000. If you carry on until our official name which is Hai'at al-Itisalat wa Taqniat al-Ma`lumat, Communications and Information Technology Commission, it will be 82 billion variants.

Of course, this one will be difficult to store. You need to have another methodology to identify all of these various labels. And we built a solution, we call it the master key. So you have one key to open all doors, you will have one key for all the domain name including the variants. And we have published a White Paper – sorry.

I really need the slides. It’s difficult now to speak. So we have a White Paper about the master key algorithm. It’s a scientific paper, and we published how we have done it and how we have solved this issue, and it is published on the Arabic domain names website, and there is a link in the presentation that you can use and see. And there was an example for three letters, [the killer] that we have shared before. You can see there are maybe

18 variants, all of them sharing the same key, and this key is used for the lookup process, so if a user wants to register a domain name in Arabic, I search for the master key, look up in the database for the master key, not the string itself, and then if it is registered already, then I say it's not available. If it is not registered, I will say, "Okay, you can register it now," and continue.

One of the important tools that we have done are filters. Filters, it's a mechanism – the goal of this filters tool is to reduce the huge size of allocatable variants. So we have seen that there are so many variants, the goal is to decrease the allocatable variants, so the variants that the user needs, and increase the blocked variants that the user doesn't need.

This was a challenge for us, so we went to linguistic and we studied the words in the Arabic language to find how we can identify the desired and allocatable variants. So we go to KAUST, King Abdullah University of Science & Technology, and we took a corpus or the repository forwards that was gathered from book, newspaper, referee, academic journals, and it has 7 million non-repeated words, and we try to use the N-gram techniques, the N-gram you separate each word to see the repetitive patters. So I need to know exactly the Alef, because Alef have so many variants in the Arabic language. So – yes, it's coming here. Yes. So this is the N-gram techniques. Okay, and then we tried to find the patters in a way that helped the user. If you want to register a domain name with Alef, then I can say, “Okay, this type are not usually good for you or it should be blocked, and these are the best options.”

We have done it, and we have put it in a ranking and weighting system, and there is a weight for each filter rule. And we found there are 21 rules, and we reviewed them with the linguistic people and they said, “Yes, linguistically, they are all right and all good.” And then we limited – and if you know someone here who can speak Arabic, he’ll know, these rules, maybe this is by default. So we cannot have Alef with Mada at the end of the word at all, or Hamza below. It doesn’t come that at all in all Arabic. It will be considered wrong name. And so on. We have built some ranking algorithms there, and we have got very good results.

This is one example. We have [inaudible] this is the holy city of Mecca. It has 3200, and using the filters and the master key, and also the international reachability I will speak about later, only six variants out of 3205. And these are the example for it, and there is a link you can go and see it all. And these are supported by this language, so the first string are used by Persian, Malawi and Pashto, the second string are used by the Urdu.

These are desired variants, the registrant can have them also. But the blocked variant is very huge. So we built our registry solution, and we call it the variant management system. It’s composed of three main things. No language mixing – and this I will give an example – and we use also the master key algorithm, and the must be allocated variant, and the filters that I told you before.

So if a user in the Arabic script world saw a sign like visitmecca.sa for [inaudible] or for a religion, and he saw the sign in a newspaper, he will use his keyboard and type this name using his keyboard. And depending on his location, he will end up with something like this. So if he's in Egypt, he’ll end up having these three code points. If he's in Saudi Arabia or in the Middle East, these three. If he's in Pakistan or Iran, he will end up with different. This we call an international reachability requirement. If you register a domain name, you need to have these variants, because user will type it in their keyboard. If you're in the airport or an Internet café in any of these countries, if you type the domain name, you will not type the one that you are used to type in your home or your laptop. No, you'll use different codepoints. And these are all bundled together and we put it in a variant management system that will help the registrant to know what are the best suitable variants and the variants that are needed for international reachability.

This is an example for the blocking quality. I have different names to the left side and the total variants, and you can see the percentage at the end of the blocked. You can see it's almost 99 point something are blocked. This shows that our variant management solutions are very good. So what we have done exactly, we believe that any registry – a TLD or even a ccTLD or gTLD – will need to support language communities within the Arabic script. Each language community should come and say, "This is my language table and code points that represent the language table, and this is the code point for variant table." Our variant management system takes each community, integrate it to one language LGR and combine them, use this as limiting the number of variants, and then generate a script LGR variant xml. We have this one, and we have a solution working, and we are now trying to promote it within the Arabic script community.

So this is an example from our registry. If someone were to register Mecca, he can say "suggest" for example, and it will list him just only the best possible choices. He can also register more if he want, but this will maybe go through human processing and to double check why he need it. But the automated will give him the best choices.

Now I finished from the tools, now I will go to the IDNA assessment reports. We have done three reports: one in 2007, one in 2010 and one in 2014, and these are links for the reports.

It shows what is the level of adoption for browsers, e-mail clients, what are the problems. Usually the problems is left to right. Arabic language is right to left, but people type it differently, the programmer doesn't handle right to left correctly, and sometimes they mix the domain with the TLD and vice versa. So these reports have very good information.

The last part of my presentation is Raseel, the Arabic e-mail. It has two phases. Phase one was 2010 until 2013, it was a pilot project test. It was built before the EAI, e-mail address internationalization RFCs, and it uses a hack. It converts the user part to xn, and the domain name of course to xn, then it sends the e-mail, and we have a plugin in Outlook that will show the user the correct representation, it will not show the xn, the ASCII representation for the user or for the domain name.

The second phase was in 2016, and still going. It's still a pilot project also. It's built on the new e-mail address internationalization RFCs, and we used Postfix, Horde, Roundcube and – this is difficult to pronounce – Archiveopteryx. I'm not sure if I pronounced it correctly, if it's not correct, it was difficult. Still beta version, it's not open for public. We have successful test internally with Gmail and Microsoft Outlook. No need for plugins, but doesn't work all the time. Sometimes it works, for example in Outlook 2016, if you use it and put it in the "to," it will work. If you put it in the CC, it will not work, or the BCC. So there are some limitation.

This is an example, [inaudible]. So our Arabic domain name – Arabic e-mail address is ready? No, it's not ready. We believe the variants that we have seen in domain names will be carried on to the user part. So if I register the user part, it needs to be protected, because you can see these two are two different e-mails with two different usernames, and two different domains. So there needs to be a protection at the user part in the Arabic, and we can use the same techniques that we have done in the domain name and inherit it or implement it in the user part.

Again, it's almost five years since the e-mail address RFCs, and the support we believe is very limited until now. Even though it's working, but it's very slow, and maybe it's support the left to right code points, but the right to left, no. It's still not supporting. There are so many bugs and so many problems.

So what is missing? From the domain part, we believe there should be automation, automated tools. So if I have Mecca as a domain name and there are three important variants I need to enable, then I need to go to the registry, register the domain name, enable the three variants or the two variants, then go to the hosting company, ask them to create from DNS point of view the CNAME or subdelegation or A record for all of them, then I go to my mail provider, have MX record for all of them and configure the user with an alias in all of them, and then I will go to the hosting company to ask them to have them aliased together. This is an example for Apache configuration, so I have virtual host and then add the domain and add the server name and alias.

So this is a problem, actually. This is a nightmare for the user. We've already seen that there are lots of variants in the Arabic domain names. So we need to have some kind of automation, maybe even new address record or new way to automate this process, because it's difficult, and ISPs don't know about Arabic domain names. They are scared, they don't accept in some cases the registrant. They say, “We'll not host your domain name.” And this is a problem, actually.

We have a gift for all registries, all registrars, all technical guys. We have summarized all of our experience in details in one huge report. Maybe it's around 50 pages. That will exact put our experience and what are the principles and things that we believe someone should take care in order to run a registry or implement something related to the Arabic domain names or Arabic e-mail address. This is the link, and you can use it.

Thank you for listening, and hopefully you can carry on our message to other parties, registries, registrars, IETF engineers to help us solve this problem. Thank you very much.

EBERHARD LISSE: I was thinking the three German umlauts, the [inaudible] were difficult. But maybe this even fits into the Chinese presentation where they're looking at provisioning ASCII and IDN names, but they also have different Chinese scripts. Maybe you can look at what they're doing, and this is helpful.

Okay, we are running a little bit late, so I'm not going to allow any questions from the floor. If you have got some time later or you can do it, take it offline. Thank you very much. And now we have Ben McIlwain from Google talking about broadening the HSTS, and he will explain to me what it is.

BEN MCILWAIN: Alright. Good afternoon, everyone. I am Ben Mcilwain, I am the Lead Engineer of Google Registry, and this afternoon I'll be talking all about security on the web. So we'll start with why HTTPS is important. Hopefully, this will be preaching to the choir, but we'll do a brief primer of this if necessary.

The main reasons that HTTPS is important are it protects users on open, public access points, man in the middle attacks. Kind of like how we’re all in this ICANN Wi-Fi right now, if you're not using HTTPS, all of your connections are vulnerable. It prevents ads and analytics injection and hijacking which is actually something that ISPs do very frequently, and it is really bad for users and for companies. It prevents spying and tracking, helps prevent against surreptitious censorship and alteration of content, promotes trust, security and privacy on the Internet, and all of these are very real concerns. These are not theoretical attacks. HTTPS is very important, and it has been a best practice for a while now for every website on the Internet to use it.

But unfortunately, HTTPS alone isn't enough. It doesn’t go deep enough through the security layers, and there are ways to get around it that can't be solved with HTTPS. For instance, there's SSL stripping, and this attack was published at least five years ago by Moxie Marlinspike where it’s very trivial if you're an active attacker that you can simply intercept any attempted HTTPS connection, downgrade it, strip the SSL and serve it up unencrypted, and that’s a problem that fundamentally, HTTPS alone cannot solve if your browser doesn’t know it’s supposed to be connecting to a secure site and it ends up getting the unencrypted version, and there's nothing you can do.

But fortunately, there's a solution to this. The solution is called HSTS, which stands for HTTP Strict Transport Security. The way this works is it’s a response header that the webserver configures, and the response header essentially means always use a secure connection, so do not use an HTTP connection, only use a secure one. So all data will be encrypted in transit. The way it works is very simple. Any attempted connection requests to that domain name are rewritten before the connection is ever made such that it is HTTPS on port 443 and must use TLS encryption in transport.

These are configured with expiration time, typically at least one year, and these can be applied at a domain, subdomain or sub-subdomain level, etc. I have an example here. apologies if it’s a little bit too small to read, but this is using some Chrome developer tools to show the interpreted status of Gmail.com, and I can't even see this from here, but somewhere on there, it says the static mode is strict. Strict is HSTS.

HSTS is very important because it prevents those SSL stripping attacks. So your browser knows that domain name is supposed to be secure and it will not connect to an insecure version of that domain. Another thing is there's all sorts of mixed content, potential attacks, or even just omissions, and that allows all sorts of data to be leaked. So your website may be served over HTTPS, but maybe you use some analytics JavaScript or something or other and that is not secure, or you loaded some third party image and that’s not secure. HSTS enforces the use of encryption on every single additional connection made within that website, so it protects these attacks. If you don’t have that, then the user’s browser can be fooled into making an insecure connection even from a page that should ostensibly secure, and that will leak information.

There's also a variety of DNS-based spoofing attacks that have been used in the real world wherein if you have control over DNS even temporarily – and things like ISPs do – then you can make a request go somewhere else. And if you don’t have HSTS, that can very trivially be a security attack. And then also, it protects against – if I'm directly typing in a URL and I don’t specify the protocol, it typically defaults to just making a request to HTTP. So even if a website is supposed to be secure, if I just type in the URL and it doesn’t have HSTS, it will first go to the insecure version of the page, and that first request can itself be intercepted and then redirected elsewhere.

And yet, even HSTS headers aren't enough for proper security, because the HSTS header has to be set by a response from the webserver. So if I'm an attacker and I can intercept the first request for a domain name, then I still have the keys to the kingdom. And basically it’s just an enhanced SSL stripping attack. So I strip the SSL and I also strip the HSTS header, and then I just man-in-the-middle the connection. So really, even HSTS is not enough for the proper security that people need to be able to do trusted connections like manage their money on a bank’s website.

Just in the news a few weeks ago was the KRACK attack n WPA2, a weakness in the WPA2 protocol which again is what we’re all using to connect to this ICANN Wi-Fi here today. Even with HSTS, if you intercept that first connection, like you haven't visited that website before on your browser, or if the HSTS header has expired – and some of the expiry times are fairly low – then you can intercept a connection and do whatever you want.

So fortunately, I'm not here to deliver a message of doom. There is a solution to even this, and this solution here actually works and delivers the full security that people probably naively expect that they're getting when they're visiting an HTTPS site but actually aren't getting. The solution is called HSTS preloading, and what this is, it’s a list of domain names that are HSTS complaint that is built into all major web browsers, so Chrome, Firefox, Internet Explorer, Edge, Safari, Opera, a bunch of others, but those are all the major ones. They all have support for the HSTS preload list, so you're all already getting the benefit of this even if you weren’t aware of it.

It’s a list of a bunch of domain names, tens or hundreds of thousands of them, and it comes by default with your browser. So before you’ve ever – say you literally just installed a new computer, you’ve never made a single connection to a website on that computer before, all of these websites are still protected. So this image is showing me querying gmail.com on this website hstspreload.org which is how you manage the list, and you'll see that it says gmail.com is preloaded. So gmail.com and the other tens of thousands of websites on the list are permanently protected from all the previously mentioned attacks because your browser will never attempt to make an insecure connection to those sites. So SSL stripping is not allowed because the browser will never see an insecure connection attempt to those websites as being valid. This gives the highest possible level of security to protect people on the web.

Now, you may wonder where I was going with this and why I'm here at ICANN, so here is a cool little trick you can do. Remember how I mentioned that you can use HSTS preloading for domain names, subdomains or anything going further down in the hierarchy? You can also use it higher up at the hierarchy. You can use it at the first level, i.e. a top-level domain. And so we did. You can see a screenshot here, this is a direct screenshot from the HSTS preload list. You can see that there are six top-level domains that are on the HSTS preload list, and all six of those happen to be top-level domains that we run at Google Registry. What this means is that every single domain name on those top-level domains gets the full level of protection of being HSTS preloaded by default without the user ever having to have successfully visited those domains names before.

There's a lot of benefit of preloading entire TLDs that you don't even get from benefiting individual domain names, which is actually kind of non-obvious and it's something that we had to work through for a while, but it's really cool. First of all, users start to associate the entire TLD as being secure, in a very real, meaningful way that a domain name is secure is more secure than even if it just uses HTTPS. So if you access a domain name on any of the aforementioned TLDs, then you know for sure that you're going to be using a secure connection and that you'll only be using a secure connection. The idea is in the future, users start recognizing these secure TLDs and start having positive associations with that.

The second one that's very important is it takes a while to add a domain name to the HSTS preload list. The typical release cycle for browsers is on the order of several months, and the preload list is built into the browser. So if I create a new website for like benswidgets.com today and I register it on the HSTS preload list, it's still going to take three to six months on average for that to roll out to all of the random web users, and that's just the built in lag time between when the list gets incorporated into browsers, when the updates get pushed out, and then when the users are actually finally getting around to updating their software. And unfortunately, users do not update their software as quickly as they usually probably should.

So if there's an entire top-level domain that’s already been on the HSTS preload list and I create a new website on that domain today, then I get protection today for all of my users rather than protection in three to six months. So that’s a very significant benefit.

And then also, HSTS headers are just kind of finnicky to configure. You need to look up the instructions for whatever webserver you're using. More modern web service [offers] better at it, but some of the other stuff doesn’t have built in support and you have to configure custom headers. So it’s just kind of tricky and it’s an additional random thing you have to do. Maybe your webhosting provider doesn’t yet support it or doesn’t make it easy. But if it’s configured on the entire top-level domain, then that doesn’t really matter so much because you’ve already got that protection anyway just by the merit of being on that top-level domain.

And then another one is it’s more lightweight for browsers. The existing HSTS list is getting kind of big, and it’s getting big in a way that doesn’t exactly scale, because there are many millions of websites on the Internet and you don’t want to have to include all millions of those on a list that is downloaded to every single user’s browser. That’s especially a problem for mobile because of memory and storage constraints. However, there are only a few thousand TLDs, so you can eliminate the need to preload lots of individual domain names, huge, vast swaths of them, if the entire TLD itself is instead preloaded. So it’s very easy to scale up to on the order of the number of TLDs, and that the list can handle easily.

Another benefit is it moves the Internet closer to the place it so desperately needs to be, which is secure by default and HTTPS everywhere. Everybody should have the benefits of privacy, security and encrypted connections. The Internet is sort of slowly going there, and it needs additional help to get there. And this is one potential mechanism.

The last benefit is this requires no custom changes or anything as far as the registry operator is concerned. It’s literally you just go to hstspreload.org and you add your domain name through there, and you can contact me and I can forward you to the right people on the Chrome team which manage that list. But there's no custom work on the registry, you're just adding it to this list that’s used by all browsers. So that’s really nice.

And then we’ll talk about some potential pitfalls. I don’t want everybody just running out of here immediately and attempting this. You need to think about it. The most important one for sure is that existing insecure websites on any TLD will cease to work, and that is a big one. Every website should be using HTTPS, but there are many that do not. And it’s probably not a good thing to retroactively break a bunch of websites with a restriction that did not exist at the time when that website was registered. So it’s much easier to add HSTS for an entire TLD that is new and that hasn’t launched yet than it is to try to retrofit it to an existing one.

Another thing is that of course, in order to use HTTPS, you need an SSL certificate, so you will require all webmasters to have those certificates and install them. Fortunately, that's a lot easier these days than it was in the past because of Let's Encrypt and others like it. And of course, you should already be using HTTPS, so if you are already using it, then there's no additional burden imposed by this.

Another potential pitfall is you can break users who are using fake local domain names on the TLD, like people before the actual TLD was released, if they were doing some fake thing a while ago with that as a pseudo-TLD, then they might break it if they try to access that in their browser. But if you were using a fake domain name that you don’t actually own, then you were kind of already broken, and especially once you hit controlled interruption, you were very broken. So don’t do that.

Another thing is that registrars will likely need special language, because this is a restriction or an enhancement on the TLD that users should be aware of, because if they aren't expecting it when they buy a domain name, it will operate differently than they are sort of expecting. Like they will need an SSL certificate. So that should be a message to them. But on the other hand, having HSTS on the entire TLD is a big security benefit. It should make these attractive to registrants because you get that immediate security rather than that sort of eventual three to six months security if you ever even do get around to the HSTS list addition.

And then finally, there are some types of sites that just cannot use HTTPS at all, like generate_204 sites. These are the sites that your browser uses to generate a captive portal page to be able to log into, say, a Wi-Fi on some hotel or plane or whatever. And those can't be HTTPS because your plane does not know how to serve a valid SSL certificate for, say, Gmail.com or something. So fortunately, there are lots of TLDs that already exist that aren't secure, and there'll always be at least some that aren't secure, so those generate_204 and similar websites can stay on there.

Okay, so our recommendations – and this is the last slide – our recommendation is if you're launching a new TLD, add it to the HSTS field list. You can join our handful of existing secure TLDs, and we can help with this. Please contact me, I'll put you in touch with the right people. And then also, any existing just normal domain names that you run, definitely get hose on the HSTS preload list. That's been a best security practice for years now. So hopefully you've already done it. If you haven't go ahead and do that. And then if you run existing TLDs with many non-secure sites, wait. There may be some additional options opening up in the future like carveouts for the list, so stay tuned.

And then if you run multi-part ccTLDs, then consider adding some of those parts to the list. Like for instance, say, you have a gov. your ccTLD code, and all the government websites are already secure. Then just add gov.whatever to the HSTS preload list and that'll secure that. And then there also may be a potential future EPP extension that'll just indicate whether or not a given TLD is secure. And stay tuned for some potential future announcements about availability of some HSTS-enabled TLDs.

Cool. So, questions?

EBERHARD LISSE: Okay, thank you very much. We have a remote question, but before we take it, I would hope that at least one of my gov.na sites was secure.

BEN MCILWAIN: You hope that one of your what?

EBERHARD LISSE: I would be very happy if at least one of the gov.na sites was secure.

BEN MCILWAIN: Sorry, I can't hear with the echo.

UNIDENTIFIED MALE: [He said if one of the gov.na sites was secure.]

BEN MCILWAIN: Yes.

EBERHARD LISSE: Remote.

UNIDENTIFIED MALE: Yes, so a remote participant has a question: “Why did you not use DNS text record to indicate that the domain supports HSTS?”

BEN MCILWAIN: Why don’t I use DNS SSAC records?

UNIDENTIFIED MALE: No, DNS text record to indicate the domain supports HSTS.

BEN MCILWAIN: So I am not on the team at Google that implements HSTS preloading. We do separately support DNSSEC on all of our TLDs, but as far as DNS text records go, yes, I think you have to be able to resolve the domain first. So if the DNS system itself is being attacked, spoofed, man-in-the-middle, which if somebody is controlling your web connections, then they can trivially do that as easily as they can intercept your web connection. So if the DNS server can be attacked, then they simply would not send that text record. So the HSTS preload list, by being built into the browser itself, as long as that has not been sort of intercepted – and it hopefully hasn't, because the binary for your browser was signed and verified – so there is actually a more real level of security that can be delivered here because the DNS itself can still be intercepted and modified, whereas your browser was not, and is still looking out to protect you even on potentially a very malignant network.

EBERHARD LISSE: Peter.

PETER KOHL: Peter Kohl, DENIC. I think there might be some TLDs that really – does it work?

BEN MCILWAIN: Yes.

PETER KOHL: Okay, thank you. There might be some TLD for which this is really applicable, and you mentioned your own as in Google ones and there are some like maybe brand TLDs and so on and so forth. Still – and I apologize, because while listening to your presentation, I was already so enthusiastic about this thing that I followed your advice, going to that website and trying to enter the TLD I work for, and I was redirected, and in the end I guess I need to write e-mail to somebody.

BEN MCILWAIN: Yes.

PETER KOHL: Which brings me to the point – so what is that vetting process or that accreditation process that would be applied to a TLD? Or could you very quickly or briefly describe what the accreditation process for other domains would look like?

BEN MCILWAIN: Yes. We've actually already gone through this at least once with another TLD operator. They're not on the list yet, but hopefully they will be soon. They will remain unnamed, but this is not just us working on this in this space, it’s sort of a – the hstspreload.org website itself is for domain names and lower, so it doesn’t actually support TLDs on it yet. So it’s still a manual addition process on our end, and it probably should be, because there's a lot of risk for getting it wrong as far as adding an entire TLD to the HSTS preload list. We want to make sure that people are aware of everything I just talked about here.

So the mechanism is – we have a public alias that is where you would send an e-mail to requesting addition on the HSTS list, and I don't know that offhand, but I can go find it –

PETER KOHL: [inaudible]

BEN MCILWAIN: Yes, so after that happens, then there is a verification process where the people who manage the HSTS preload list verify that the people they're being contacted by are the people who actually run that TLD. So I think they're doing something like they're looking for the registered administrative or technical contact for the TLD according to something in ICANN’s database or something, and if they get an e-mail from that e-mail address, then they’ll have some confidence that they're actually talking to people who have the rights to make this request.

PETER KOHL: Thank you for that explanation. Would you be aware of a mechanism to have an explicit preemptive nonentry on the list?

BEN MCILWAIN: Interesting. So like basically being contacted by a TLD owner saying, “I am who I say I am, and we do not want to be on the list. Please don’t add us to the list.”

PETER KOHL: Yes, exactly.

BEN MCILWAIN: Yes, I'm not aware of anything like that. It might be worth considering.

PETER KOHL: Thank you.

EBERHARD LISSE: Okay. I'm not taking any more questions because I'm running a little bit late, but you can contact him directly. The last presentation is from – thank you very much – CNNIC about how to bundle names in registrations in IDN and ASCII [TLDs].

NING KONG: Hello, everyone. I think we are running out of time, so I'll try to introduce my presentation as brief –

EBERHARD LISSE: No, you have your 20 minutes if you need them.

NING KONG: 20 minutes?

EBERHARD LISSE: If you need them.

NING KONG: Okay. I think maybe 10 minutes is okay. I want to use some time to really talk about the resolution of bundled names.

So what is bundled name? From my mind, we have several use cases. The first one is under the same TLD and we have the second level domain names at the variant label. So for example, we have English variant and the British style for "colour" and the

American style “color.” If someone register “color” and they're a .com, and maybe they want to both register the two kinds of “color” characters within .com, and they want to mange it by themselves and [inaudible] the same purpose.

And for the Chinese domain name – and we have the simplified Chinese character and the traditional ones. So for CNNIC, we have the registration policy if one registered regional Chinese character domain names and they will get pure simplified Chinese domain name and the traditional ones, so that means anyone want to register one Chinese domain name, they will get multiple Chinese domain names, not only one. And for Chinese users, the simplified Chinese domain name and the traditional ones are totally the same.

And for the second use case, we can have the same label for the second level domain names with different TLDs. For example, maybe we can register example.com and example.net, and maybe I can registered ning.com and ning.net and ning.cn, and I want to use all of my names from different TLDs used the same ways. I will register all kinds of my names under different TLDs, and one users to access any kind of my names under any TLDs and will access my website, maybe.

And for another example is a PIR, and they manage the .ong and .ngo, and based on their registration policy, anyone register one name from PIR, they will get two domain names, for example, example.ong and example.ngo.

Another use case shared from .gr – I'm not familiar with the Greece character, I just copied someone from .gr, and they also have the IDN brands. And for example they have the different sigma. You can see there are three kind of IDN brand, and some domain name will end at normal sigma, and some ends at final sigma, and some other names have the tonos and some do not have. So that means from .gr – and another presentation just to [inaudible] presentation .ascii and they have another IDN brands.

So another use case from GoDaddy. GoDaddy shares today, and people will register only one domain name, and they will set up the website, use one domain names but use different purpose. So maybe in the future, people will register different domain names with different new gTLDs, but such kind of bundled names, we think the people who want to register a bundled domain names but each domain name is maybe used a different purpose.

So what are the requirements for the resolution of bundled domain names? We want to talk about the bundled domain names and the registrant want to have the same purpose for the whole bundle of domain names, not – we do not want to discuss the bundled domain names with one registrant but used different purpose. So maybe we can have t wo ways of requirements. For the option one, maybe we want to only allow the registrant to activate a finite number of the domain names based on the .sa's presentation and the IDN brands. If the label is long and the number of variants could be huge, so if we want to choose the option two, that means we want to make infinite variant numbers of domain names to all can be activated, all can be resolved, so we think it's another big challenge.

So if we want to have a technical mechanism to allow the registrant who has a bundle of domain names and want to resolve them in the same way, and maybe we can have several technical solutions, but I think the current technical mechanisms are not perfect. For example, now we have CNAME record, but CNAME can only allow us to map the domain name itself. We cannot use CNAME to map the second level or the children levels.

But for the DNAME, DNAME only mapping its children, not mapping itself. And I know some registry now use some out of band solution, maybe provide some service system to the registrant to manually configure ways to [allow] the registrant to set up the bundle of domain names to resolve in the same purpose. But we think the out of band solution maybe is not very efficient, and maybe we'll have a manual error.

And maybe in the future, we can have more in-band solution of DNS, and someone [from] my colleague propose ways. Maybe we can create a new record named maybe bundled names, BNAMEs, and BNAME can allow the domain name mapping not only itself but also its children. But if we have the new record, that means we need to let all the resolvers to know it, and it’s a big challenge.

Another possible solution is the zone clone. Someone, Paul Vixie proposed such technical solution, and they want to copy any DNS zones from one domain name from the bundle domain names and to each other’s DNS zones. But this solution has a problem. If there is a children be delegated to other ones, so it’s hard to copy the whole zone files to another.

And another solution, I think this was proposed by CCNIC, and from this technical solution, people are allowed to manually configure CNAME plus DNAMEs. And we think such kind of solution also has the problem that all of the resolvers must support such kind of new features.

So general questions, I'm just wondering for the ICANN or for IETF, do people need to think about to provide some kind of solution to make the bundled domain names to be resolved in the same address? And if so, how can we do it? We try to modify the DNS protocol. We do issue this problem in the IETF community, and I understand a lot of DNS experts think that it's not DNS, it's [sales] problem, it's maybe application layer problem.

So someone proposed that this is a problem, but is complicated, and we should not only think about to change the DNS, and it's not enough. If you only change DNS layer but application – you cannot make sure the application can make the bundled domain names to be used as same way.

So if we do not [change] the DNS ways, we think some new things, maybe we need to create a new layer above the DNS layer and we create a new fundamental service to make some kind of mapping service, and from this new layer, this layer can judge if one domain name's been bundled with other ones based on maybe IDN table or the bundled name tables.

But I think another new challenge is, how can we make such a fundamental service just like DNS to make effect in the near future? And also, I think that maybe we should think about, do we need a mechanism not only in the naming system layer but also the application layer? Even we make sure that bundled domain names have the same address, but we cannot make sure the application to look the different bundled domain names [are] the same one.

So the questions I want to propose to the attendees, and do you think this is a problem ICANN community should think about? If so, ICANN should be thinking about the problem statement of the bundled domain name resolution, policy issues from ICANN, or maybe in the future, we need to do some new works in IETF? And if so, what kind of company or organization do you think should be involved in this problem? Any gist from the room? That's my presentation. Any questions?

EBERHARD LISSE: Thank you very much. Something I've never thought about other than that we may be using Levenshtein distance to find misspellings where we can relatively easy do this. But bundling, I have never looked at it. Maybe it's something that you can solve on the registry level [that the] registry then basically CNAMEs or DNAMEs, or does these things together.

One question – and I want to see, do I see Norm Ritchie here somewhere? Oh, he escaped. So Filip, can you then do the closing like you promised on [inaudible]. [inaudible] has the last question.

ROBERT MARTIN-LEGÉNE: Hi, I'm Robert Martin-Legéne from PCH. It's not so much a question, it's more in terms of support of – I think this is the right forum to ask, because the bundling question is mostly a registry issue, I think. Each registry sets policies for what is allowed and stuff like that, so I hope that everybody who has these problems will find together. And I think there's a little bit of a vague momentum right now, but I think it’s a real issue that you have, and somebody should do something as to where to solve it. I am not sure, to be honest.

EBERHARD LISSE: Okay, thank you very much.

UNIDENTIFIED MALE: Yes, thank you.

NING KONG: Excuse me, I share this topic with IETF guys, and some IETF guy think the IDN variance problem is a swamp. People think it’s a problem, but it’s too complicated, maybe cannot fix it perfectly. Thanks.

EBERHARD LISSE: Thank you very much. As usual, we have somebody who didn't present who is a member of the Tech Working Group to give us a little overview what he thought about it or she thought about it, and Ondrej Filip will do it since Norm Ritchie managed to escape.

ONDREJ FILIP: And of course, I wasn't prepared for that, so I will just have a few not very structured thoughts about this day. We had again very beautiful meeting today. For me which was really positive that we saw a lot of people from this region, and that's important. I hope that those guys – who were very great, and thank you very much for coming, guys – they'll follow those meetings in the future, because such a good presentation we have every ICANN meeting, so please come and share some ideas from your region with us and try to learn from our mistakes, of course.

Also, something that was scary for me was that before this day, I thought that my language is complicated, and then I realized that this part of the world has completely different problems, problem on completely different magnitudes of scale. So that was something scary for me, and no, I'm not sure how you can use your language with your computers here on the Internet. You're very brave guys, and you face a lot of challenges here. So that was great.

And also what was for me a highlight of this day – and I hope we will keep those big ideas coming, [this meeting was a presentation from Jacques.] I know it's not easy to come with such a broad idea and try to sell it. Everybody saw that there are millions of issues, but if we will not try to overcome those issues, if we'll not try to make this happen, it will not happen, of course. So let's break your idea into some small steps and try to go forward. I think the motivation behind it is pretty obvious, to keep the relevancy of the registries, and that's probably that many of us are facing these days. We cannot survive just by the growing number of domain name registrations, because that's not happening these days. We are in the face of stagnation, so trying to find some other value, find not for us but for the community, bringing some added values to community would probably keep us in the business. So I hope we'll see more such ideas on tech days, and I think it's a great forum for exchanging those ideas.

That was my last thought. Let me thank Eberhard for running this excellent show today and thank him for creating the program. With that, thank you very much for coming, and I hope I'll see you all in Puerto Rico.

EBERHARD LISSE: You most certainly will see me. I have already booked.

[END OF TRANSCRIPTION]