BARCELONA – Tech Day (4 of 4)
Monday, October 22, 2018 – 15:15 to 16:45 CEST
ICANN63 | Barcelona, Spain

EBERHARD LISSE: Okay. Come sit down, guys, and we can start. The next presentation is about an open source HSM, which is one of the topics that I'm most interested in because I feel that DNSSEC is easy to do, but difficult to get audited. And expensive.

RUSS HOUSLEY: Good afternoon. I'm Russ Housley and I hope you find this topic interesting. So, this definition of what a hardware security module is comes from Wikipedia. It's basically a hardware device that safeguards and manages encryption keys that are used for strong authentication or other cryptographic processing. These modules often come in the form of a plug-in card or an external device that attaches directly to a computer or server.

So, why do people care about these things? They're basically a lockbox for your private keys that are used for DNSSEC or RPKI or Web PKI or the TOR network for onion routing or corporate authentication. Other people use them for encryption and decryption, virtual private networks.

In order to do cryptography well, you need a source of random numbers, so some people actually use these just as random number sources.

They come in all kinds of different flavors and sizes. Some of them small, what I call personal portable crypto tokens, something you carry around in your pocket. Others are permanently mounted in racks, so that they don't walk away. But, can you trust what's inside? Well, there's been plenty of examples where there were surprises, even though they had been evaluated by governments.

So, it seems that every week we're hearing about some new surprise, some more horrific than others, through compromise, or malware, or back doors. Yet, we rely on them to keep our cryptography safe. So, several of us were not comfortable with this, just because it's inside this container, we have to trust it.

This cartoon of the press asking two national leaders, "Do you have any comments on your outrageous cyber spying?" And neither one could decide of the question is for them or for the other guy. It summarizes the political environment that this all is going on in.

So, this effort started because myself and Jari Arkko, who was the IETF chair at the time and Stephen Farrell who was the security area director at the time, observed that we really needed an open and transparent way to address this, but I want to make clear that this is not an IETF project, it's not an Internet Society project, although both have contributed in various ways. But, there were people who care about the security of the Internet.

So, the goal is to provide an open source reference design for the hardware. We're not manufacturing it, for the most part. The idea is that you manufacture it yourself or you can buy one that somebody else has

manufactured, but the project itself is not trying to do production scale manufacturing, although we have built some prototypes to make sure it all works.

Scalable [inaudible]. It's got an FPGA and a CPU in it. Maybe higher-speed things will come later. But, the idea is that you get assurance because the design is open. There's a diverse development team each looking over each other's shoulders, and over time, we hope to increase the assurance of the [tool chain] as well.

But, that's the goal. Give me something where I can look at the hardware. Give me something where I can look at the software and have other people looking at it, too.

So, I already said it's not a product, but it's open. Everything is open. The documentation, the design, the code. It's all under BSD and CC license. As I said, it's a diverse development team. All of this is for transparency. The testing is all readily available to everyone. And the funding source is diverse, as well. We actually purposefully put a cap that no one person or no one organization could become the 800-pound gorilla from the funding perspective and we had to do what they wanted. That was just another way of making sure that it was open.

These are the sources of the people in terms of who are doing the work, the design hardware, the code, the FBGA code then, the CPU code. As you can see, they're from all over the place. That was intentional.

So, the first practical implementation is the DNS signer. It works with openDNSSEC, BIND, [Knot], PowerDNS. It supports both RSA and DSA.

And a company called Diamond Key Security is basically offering a [cryptech inside] kind of a product. They are also making all of the code for the thing they're putting around it available, and some stuff we're working on right now is in final testing or [hash-based] signatures which are a kind of signature that is quantum safe, and under development IRTF Crypto Forum Research Group has chosen for the next generation traditional scheme a digital signature called ED25519 and that's under development right now. We expect it to be code complete and fully tested by the end of the year.

This is kind of a picture of how it's all working. Just for the more technical people to understand how all the pieces go in, the point is that the bottom pieces of this are the ones that are a part of the project and the Diamond Key, for example, offers a tamper-protected boundary around the chips and so on. I'll show you a little more about that later.

This is what the board looks like in terms of the ones we built for the developers to use to make sure it was all working and have had a couple workshops where people have tried to use Standout DNSSEC, having the keys generated on the board and stored in the board. That's one way we made sure it works with all that code.

For those who like to know about the hardware details, it's all here, but the bottom line is it's all available in a repository. By the way, I don't think I put the URL on the slide. It's cryptech.is. The reason it lives in Iceland is they had the best privacy laws to ensure the repository wasn't taken down.

So, what did we get done this year? We made performance improvements. We're basically revisiting and updating the implementation to improve performance. Some of the work has also led to improved security about migrating some things that were done in the CPU into the FPGA, so that the key material stays in one place. The hash-based signatures, which is based on, as I told you, the IRTF document. I gave you the URL for that there.

The big thing we think that will be used for is code singing. The situation is if you have a quantum computer come along, how do you deploy the next generation crypto if it's signed with RSA or ECDSA which are vulnerable to quantum computers? The idea is let's start signing now with something that is quantum safe and that way, when we have to deploy the next generation, we have a way to do so in a secure manner.

As I said earlier, we're working on wrapping up the ED25519 and another thing we did this year is we had Cure53 do an external audit of all of the code and design. In following our open source model, we posted their report on the website. We had identified a couple of vulnerabilities and we expect to have those fixed by the end of the year.

Looking at next year, we're hoping to update the hardware, continue to improve security, and continue to make performance improvements. We do want to make sure that we follow along what Diamond Key is doing and someone building a product around this has all the pieces they need and input for new use cases is always solicited. We'll certainly put it into the hopper and maybe it won't make it next year, but maybe the year after. Those kinds of things, depending on funding.

At this point, this project has been just under $2 million and we've got about $200K left in the donations we have. So, of course, additional contributions are always welcome as well. These are the guys who have sent funds so far and we appreciate that.

For those of you who care more about the technical details, a backup slide is in the deck. I don't intend to go through them. I don't think we [inaudible]. I'd rather have questions.

EBERHARD LISSE: Any questions from the floor? Warren? It would be so much better if we could hear you.

WARREN KUMARI: Yeah. Oh, this one even works! So, Russ …

EBERHARD LISSE: But we can't hear you very well.

WARREN KUMARI: So, I like the idea of not trusting one of the well-known vendors. But, why should I trust you?

EBERHARD LISSE: Because you are funding it.

RUSS HOUSLEY: So, you shouldn't trust me and that's the whole point. That's why the diverse design and code team, we very much want each other looking over the shoulders to make sure that no one person or even two people can be doing something even subtle, but the idea is that anyone can take a peek and further the independent review by folks who didn't write any of the code. It did help us find some vulnerabilities and we intend to continue to do that. So, hey, do a code review if you want. Tell us what we screwed up.

EBERHARD LISSE: Come on, stand up. If you have questions, come to the middle. It's easier than …

UNIDENTIFIED MALE: [inaudible] from dot-PK registry For Pakistan.

EBERHARD LISSE: You are difficult to hear.

UNIDENTIFIED MALE: Okay. [inaudible] from dot-PK registry from Pakistan. So, why not open up the reference design and anybody could put it together? Because if I get a box, I'm always like, "I don't know what's inside the box." [inaudible] becoming so small. I put a small [IOT] chip transmitting whatever to wherever. So, is that possible? Is that something that maybe [inaudible]?

RUSS HOUSLEY: So, we have a couple of different things going on there. You could go to buy your own parts and get out your soldering gun and do it yourself. The TOR project is taking the design and the form factor doesn't work for them, so they're building a PCI board that's going to fit in the nine servers that do the core of that system and they are basically taking the design, taking the code, putting it on a different form factor, but essentially all the same parts, reusing all the code, and doing a code review on top of it.

So, I'm thrilled about that one because it's an application where some other people have a need and they're willing to pave into the community, so I'm excited about that one. So, it works both ways. Build your own or take the design we have and have it manufactured. Either way. Hi, John.

JOHN LAPRISE: Hi, Russ. I going to ask sort of the same question, which is considering some of the stuff that we have seen reported by Bloomberg in the past couple of weeks, do you have any process for verifying that the board as built is the same as the board as designed?

The other question is I'm just wondering, do you have any feeling for who is likely to be the people who use these things? As one of the people funding Diamond Key, I believe there are such people. I'm wondering who are their customers?

RUSS HOUSLEY: So, I'll let Diamond Key speak to who their customers are because I'm not part of Diamond Key. But what they have built is a rack mountable device with an Ethernet interface so that it can be used in DNSSEC applications that live in a data center, so that's their target. It's got the tamper wrapper and so on. So, the Bloomberg thing. There's certainly reports on both sides of that one.

JOHN LAPRISE: I realize that everyone's denied it.

RUSS HOUSLEY: Exactly.

JOHN LAPRISE: But it has that aroma of plausibility.

RUSS HOUSLEY: So, it is certainly a possibility that in the manufacturing of the hardware process, someone could add something that the designer did not intend. So, we are not trying to solve the supply chain problem. That's an interesting problem. You could build your own. You can have them built by two sources and compare the difference. There's several approaches you could take. So far, we have had two different fabs, but both, so far, have used the same [inaudible] circuit board.

EBERHARD LISSE: Okay. I would like one more question. Frederico?

FREDERICO: Frederico from nic.br. Russ, in the same lines as the earlier question, how can we trust this new black box, even not [inaudible] two internal black boxes that are inside of it? The [inaudible] and the FBG. These are [inaudible] in the …?

RUSS HOUSLEY: Again, this is a supply chain problem. The tool chain is part of that. If I'm going to build some components, how far do I go down? I can't afford to build my own CPUs and build my own FPGAs and so on. You basically have to draw a line at some point and say I'm accepting this amount of risk. Your mileage may vary, but I would argue that having to trust, in this case, the arm chip and the FPGA from [inaudible] log is a significantly different thing than buying a box that's got all the [inaudible] protection around it and you have absolutely no idea what's inside the skin. I think the open source, having multiple eyes on it model is giving you some assurance but not total assurance.

EBERHARD LISSE: Okie-doke. Thank you very much. Next presenter is Patrik Falstrom, I would think, or whoever is here from SSAC or [inaudible] I see there about the IDN homograph attack that there is [inaudible]. Oh, we have more presenters than one. Even better. The more, the merrier.

UNIDENTIFIED MALE: We have ourselves a panel.

EBERHARD LISSE: Ladies first.

ROD MASMUSSEN: Hi, good afternoon, everybody. Rod Rasmussen, [inaudible], Tim April are all here from SSAC. We wanted to talk to you today about some interesting developments in the use of IDN homographs in a series of attacks that occurred over the last several months. This is something we've talked about in being a theoretical threat. It's now becoming much more of a reality on the Internet.

Really quick background. We're SSAC, the Security and Stability Advisory Committee. We've got just shy of 40 members. We publish advisories to ICANN around SSR issues and look for emerging threats. We're trying to do a series of presentations on emerging threats at Tech Day every time and we look forward to briefing you on a regular basis.

So, we're going to talk about … Well, this is the agenda. We'll talk about IDNs, Unicode and DNS labels, what an IDN homograph is, detecting and mitigating them and some ideas around things that can be done in this area and then hopefully have some Q&A.

We also have Suzanne Woolf listed. She is unable to show up. Oh, she's here. Okay. Come on up, Suzanne. And if you weren't prepared, I'll take care of it. We also want to acknowledge that this data that we're presenting was developed largely by Mike Schiffman at Farsight Security and also [Sam Herb] at Akamai and it's been valuable for

shining a light on some of what is going on out there. I'm going to turn it over to Tim.

TIM APRIL: Just to give you a little bit of background or anyone that doesn't know what an IDN is. IDNs are essentially the way of making it so that, in the DNS, you can have things that are not just the basic Latin character set. So, you can have something for whatever geography or whatever language you're normally speaking.

The IDN 2008 is the current standard for how IDN is mostly defined. It goes over how the translation will happen and I'll talk a little bit about that in the next couple slides. But, like we said before, it lets you use non-Latin or non- basic Latin characters in the labels of DNS. So, you can either have the entire domain name – so, both top-level domain and any of the sub-domains under it that are in, at most, one other script type. So, you can have Russian. You can have a second-level domain that's [inaudible] Korean or some other language with [inaudible] dot-com. And then you can also use languages that have different display properties. So, if you wanted to use Arabic, you can have it go right to left. Where, in this case, the TLD is actually – in the Arabic example, the TLD is all the way over on the left side of the display. That leads to some interesting issues when you have an Arabic label and then you have some other script that's written in the other direction.

So, some basic terminology for the things we're talking about today. This isn't a complete list of definitions but this is what was available in

RFC 63, 65. A language is just the way humans communicate. A script is a graphic representation of characters in its written form. And then a writing system is the rules and conventions of how you operate in a particular language.

The things specifically related to this talk, a character is just a single icon in a writing system. So, you can have an A and an A can be represented in different ways, whether you're using Arial or Helvetica or something like that. And each of those representations is a glyph. The slides are just being weird for me.

Inside of a label, you can have multiple glyphs. The idea of a homoglyph is where you have two representations of a character that look similar. So, in this case, you see the two As where there's the tilde A over one A. Those would often be seen as a homoglyph, because to many people, they look the same. If it's your native language, you may have a better understanding of how that will – the difference between the two. But for people unfamiliar with it, they may think it's the same and it can be confusing.

Then, a graph is one or more glyphs that make up a string and a homograph is where one of two or more strings appear identical or very similar. So, in this case, there's the same A with a tilde over it and those two labels are homographs of each other. Or often considered homographs of each other.

One thing that didn't make it into the slides as we were experimenting right before the talk and we were actually able to register a homograph of icann.org – if you come up to me later, I have it printed out if you want

to see what one looks like that we found the other day and were able to register.

Unicode and DNS labels. Unicode is a single character set that is able to display many different languages. The goal of Unicode is to display all of the written languages that they are able to. Inside of the character set, there are what we call code points, which is representing a single character.

So, you can see at the beginning and the middle half of this slide, there's the basic Latin characters which is often referred to as ASCII where they're represented in the Unicode format where they're very early in the phase, or in the description.

There are also many other languages, such as Russian and I believe that's Chinese. I can't remember. Chinese or Korean. I'm bad with languages. It also can include other things like emoji and math symbols. I don't want to get into the emoji and DNS labels at this point. That's an entirely different conversation and I'll wait for Patrik to give that one.

So, when you bring Unicode into DNS labels, this is where we have to talk about the difference between a U-label and an A-label. When you're using an application like a web browser or some smart board application, you may see unicode.com or something like that where that representation is what we refer to as the U-label. When your machine actually does the lookup, it could take all of those Unicode characters, and using the Unicode string representation, ask the DNS server, because that's technically allowed. With the DNS protocol, you can put any bytes you want in there. But, the convention is to use the A-

label, which is where you have the XN - - in the front. And that's the representation that goes across the wire because most registrars won't let you register anything that isn't an ASCII character string. Or at least that's now how it appears on the backend.

And when you do that transition, you end up with the XN – - that's a one-to-one mapping between U-labels and A-labels and that encoding is often referred to as Punicode. I'm going to hand it over to [Metica] to talk about the attacks we've seen.

UNIDENTIFIED FEMALE: So, also to reiterate that there's also a lot of ASCII lookalikes. So, ASCII lookalikes is, I think most of you in the room are already aware two more ASCII strings that appear identical are very similar. There are solutions that exist for detecting some ASCII lookalikes, but they do not exist for IDN homographs and that's really something to pay attention to.

Now, there are homographic attacks that exist. I'm not going to read through the slides, because I'd rather talk you through some of the data. And here's some of the examples, just to show that they do exist. The key point, also, is that they exist for many, many, many brands. When we look at observations in the wild, there was a talk at DEF CON from Akamai where business it showed that you had close to 2,000 impersonation domains observed using the certificate transparency. And I think most of you know what a certificate transparency is. But, for those that may not, it's an open framework for monitoring SSL TLS certificate systems and being able to audit specific TLS SSL certificates.

The effort started in 2012. It was done through open specifications with the IETF and also private industry.

Also, there was Farsight Security research that was done in January 2018. Some of you may remember. I actually gave a talk on this in ICANN 61. I was working at the company at the time. At that time, the research that was done by Mike Schiffman, one of the engineers, he examined 125 brand names and in a three-month period observed over 116,000 homographs. In addition, he was just kind of curious and he looked at whether or not they had websites. Many of them did. What was even more interesting, some of them had live phishing sites. So, we had done some responsible vulnerability disclosure, actually contacted the brand names to let them know that this existed, so they could take a look.

But, I want to be very clear, that even though Farsight sees a lot of these homographs, they don't necessarily look at all of them to see whether or not it's actually used for malicious purposes. They're suspicious. And I want to make that very clear because some of these may be quite legitimate. That's also something to really take a note of.

But, Farsight has continued to look at these trends. So, the research has continued. So, now, they've examined up to 509 brands and in a 20-month period observed 11,000 unique IDN homographs. And you can see some of the numbers. Basically, they observed 61,000 total IDNs and in varying different sectors.

So, here's some graphs that I think some of you may find really interesting. Over the entire time, Mike has done graphs from January of

2017 to October of 2018 and basically he's seen 161 million total IDN observations, 34 million total unique IDNs. It's interesting to note that I guess people also take the summer off, because from June to August, you see there was a great dip. So, not sure why, but it exists, at least from the data observed.

Also, you take a look at some of the trends and you looked at the top 10 IDN TLDs. So, in total, there were 1,675 total unique top-level domains and this here shows the top 10.

Also, when you look at the total observations of total IDN homograph observations, there's 11,766 total unique IDN homographs. What this particular graph shows, the trend is increasing, which I think is something really interesting. I saw this talk at the DNS OARC meeting about a week ago and Leslie Daigle who used to be the chair of the IETF said we've been discussing this for so many years. She was the chair when IDNA 2003 came up and it was interesting because, even though it was discussed, this actually shows that this is actually happening.

So, this is a slide that shows the varying sectors. Basically, it happens in every sector. So, getting to detection and mitigation, Rod?

ROD MASMUSSEN: That's Suzanne, [inaudible]. Okay.

SUZANNE WOOLF: Yeah. I can run through these. Basically, you're relying on all of us for questions and comments. Next, please. Alrighty.

So, the question becomes now that this has actually been seen in the field and so on, obviously there's a higher level of urgency as far as how to detect attacks and how to maybe even prevent them.

Detecting attacks, it's a matter of vigilance. Monitoring certificate transparency logs, monitoring DNS zone files. [Passive] DNS services can be extremely helpful in detecting IDN homographs reliably because it's somewhat a matter of judgment and context does require a human in the loop.

But, as far as mitigation, and frankly, as far as being able to prevent some of the potential problems. One place to put some of that is stricter rules of the registry and registrar where domain names are generated and the problematic ones perhaps can be prevented from being created. There are recommendations in IDNA 2008 which explicitly recognizes that the standard can only define the set of characters that may be acceptable in certain contexts. They can't decide for you what's correct for your user base, for your communities of interest.

Using an inclusion-based process before allowing code points just means, not assuming that you'll use every available or eligible character unless there's a reason not to. It means starting only by using the ones you're sure are safe and will be useful to your community of interest.

The suggestion to be extremely conservative with mixed scripts within a label and within a domain name, that's also something that can be detected and when the attempt is made to create the names, that can

be very difficult to find later. Adapt the label generation rules that ICANN has been working on. The basic vocabularies for different scripts.

Browsers often implement homograph preventions as well, but because it's at a different place in the evaluation process of looking at a URL and sending a user to a URL, that can be actually more limited than being able to examine domain names at the registry and the registrar.

Just to emphasize why this is important, the mission here of security, stability, and resiliency of the global unique identifiers in the Internet. We're trying to prevent further encroachment by phishing, malware, malicious e-mail, all of the familiar things we guard against. Because it undermines trust, failing to take care of some of these challenges affects universal acceptance. It makes it more difficult for browsers and other application operators to accept names and URLs that they see, which may result in ad hoc or overly broad blocking of names or refusal to display certain things.

Business e-mail compromise is a growing problem anywhere and everywhere. Failure to act, again. People will do what they have to do to protect themselves and their users, including just blocking anything that looks like it could possibly be suspicious.

So, what the broader community can do, there is the opportunity for development of tools to detect IDN homographs. Comparison to known targets, visualization, the ability to facilitate brand protection is one of the drivers for this.

Awareness and outreach are the potential malicious use of IDN. What we're trying to do here today, but also what you can take from this. Just more awareness that this could be a problem, that this has become an actual problem. End user awareness. Implementer education for people that are writing tools and applications that use domain names directly. And service provider awareness. And we will hand it back to go through any additional comments and hopefully questions.

ROD MASMUSSEN: Okay. Here's a quick list of relevant SSAC publications that touch on many of these things. As you can see, we've got five different publications – several of them are fairly recent – that take a look at IDNs and their use and guidelines, etc.

I want to emphasize a point that may get lost in going through this presentation to make sure we have time for questions. There are lots of tools out there right now to find lookalike domains. There were some examples given earlier. But a good example is if I put a zero instead of an O in writing Facebook or two zeroes instead of two Os. There are lots of tools that will let you find that because there's a [inaudible] mapping.

The challenge with IDN homographs is that it takes the visualization within a script in order to be able to see that. It's not a direct easy mapping to do on a consistent basis and there's definitely a lack of tools and capabilities out there, so that's one of the areas we see as this problem grows and it will continue to grow as long as it's effective. That's a big need for us to be able to detect this stuff through visualization and that kind of an approach. I want to make sure that was

emphasized. And one of the reasons we're bringing this up today, not just the fact that, hey, we're finally seeing this. We've been talking about it for years. Probably over a decade we've been talking about the possibility, but now we're actually seeing it, and we don't have effective tools, necessarily, and effective prevention mechanisms from stopping it from growing today.

So, with that, we'll turn it over to any questions people have.

EBERHARD LISSE: Any questions? I must say that we see not only that the [inaudible] on vacation. They also don't really like to work weekends. From the floor.

ROB GOLDING: Hi, Rob Golding, [HTATM]. Are there any tools which can put PayPal up in 47 different fonts and then put PayPal up in XN - - blah, blah and tell you it's an 82% likely match of something dodgey?

[TIM APRIL]: There are some tools out there that aren't generally available. I know some institutions have developed their own method where they do some sort of comparison where they render it in a set of fonts and try and see if they look alike. But right now it mostly comes down to a human. There's a tool that will put a web page with 100 different fonts and the actual label next to it in a couple different fonts. You look and see, "Okay, that looks alike." But, a lot of it comes down to what font size do you render it at? What font do you render it? And whether or not

the tool you're using will actually display that representation of the string.

ROB GOLDING: And is there a possibility of perhaps pre-generating all of those so that you can provide a list and say, "If you get asked to register bank, it looks like this in every possible script, in every possible language."

TIM APRIL: It's possible …

ROB GOLDING: But, it's [inaudible] something that's being looked into at the moment.

TIM APRIL: I know it's being looked into by some institutions but not as a well-formed approach.

ROB GOLDING: Okay, thank you.

ROD MASMUSSEN: And that's one of the things we wanted to bring to folks attention is this is an area we need more research and more tools being built.

UNIDENTIFIED MALE: Hi. I'm some random guy who you don't know. I guess I have two comments. One is I think we may need to start being really, really cruel and hard-assed about what we consider to be valid labels. For example, Tim's thing that looks like icann.org is in fact mixed script because the org is in Latin and the ICANN is in Cyrillic. There's a lot of people who try to do branding. At some point, we may need to just say, "No, forget it. It's all got to be the same script all the way from one end to the other." I imagine socializing that here at ICANN will be a challenge.

But, the other thing I was going to ability is trying to kill homographs, you can't. There's too many ways to write things that look like something. To what extent do we need to go and look at the stuff in the other direction about how to figure out how to put a gold star and say, "No, this really is your bank."

Traditionally, it was supposed to be certificates and then it was supposed to be [EB] certificates and now there's other things like this thing called [Bimmy].

But, I'm wondering, to what extent should we look at that and try to point to things that might actually work.

SUZANNE WOOLF: Just to pick up on that point a little bit, that's kind of the idea behind doing inclusion-based [rapporteurs] of acceptable characters and that can really only happen with the registrar and the registry. It is a matter of saying not every possible valid label is a good idea.

UNIDENTIFIED MALE: If you look at the LGR list at IANA, it already makes me very sad.

SUZANNE WOOLF: But also, there's always the compromise there.

ANDREI KOLESNIKOV: I just want to echo that the reasonable tool which prevents from the homographic attacks is a restriction of the labels in [neo] script. The statics on [dot-RF] and I kind of can guess why it's there is because people do like [www.putin.rf] in Cyrillic. It may look like mixing script, but in the second-level domain, there is nothing but certain Cyrillic table in the registry and you cannot register any other label in the zone. So, it may look big because it is big by the fact it's the largest IDN.

But, what the guys collected for their log analysis, basically, I think is just [inaudible] because a lot of bots generate automatic brand blah-blah-blah something dot-RF and trying to resolve it or doing something like that. But it's simply because in the dot-RF there is no ASCII. It's just Cyrillic on the left and on the right. And I think absolutely right. The only way to prevent these homographic attacks is just to have a straight label generated restriction in the zone. Thank you.

UNIDENTIFIED FEMALE: Yeah. I'll just comment that you're correct. The data that we showed from Farsight was actually using the passive DNS which is observations from above where the curser—

UNIDENTIFIED MALE: [inaudible] goes there, right?

EBERHARD LISSE: Last question.

BARRY LEIBA: This is Barry Leiba. I'm responding to John's statement that we should instead find a way to tell people that they've got where they really wanted to go or they have what they expected.

This came up yesterday and I had the same thing to say. We techies have this idea that telling users what's going on is a good thing. Study after study shows that users don't understand what we tell them. We cannot rely on telling users that they've gotten where they're supposed to go. We can try, but it just isn't sufficient.

EBERHARD LISSE: Personally, I think it's not so much that they don't understand. It's that they don't care. There are three billion people and you will reach 2500 who perhaps understand it and maybe …

UNIDENTIFIED MALE: There is some of the "don't care" but, for instance, studies have shown that the lock symbol in the browser Chrome is confused by users. They think that if you put a lock symbol on the web page content it's just as good. The users don't understand these things we're trying to convey to them and this is why techies should not design user interfaces or try.

EBERHARD LISSE: My view on this is if you invent something, somebody will monetize it. Thank you very much. Okay, Geoff Huston?

GEOFF HUSTON: Good afternoon, all. My name is Geoff Huston. I work with APNIC and I do a huge amount of measurement. One of the things that's happened in the last … Geez. It was October the 11th. A little over a week ago now. We finally managed to do one of the, I suppose, trickiest things we've ever tried with DNSSEC in the DNS. That's actually roll the top-level key signing key.

The reason why this is tricky is that inside a hierarchy of keys, down in the hierarchy there's always someone above you. So, if you need to roll a key, you tell your parent the new key. Your parent then has two keys and then after a while, when everyone has learned both keys, you retire the old key. That's great. Unless you're at the top of the apex.

When you're at the top of the tree, you've got a problem. The only reason why we trust this KSK is because your validating DNS resolvers have loaded this key value into that code.

So, how do you roll it? Literally we need to change the trust point for every validating DNS resolver. That's hard. What's made it a little bit easier is an automated process. It's defined by the IETF in a standard called RFC 5011 and it relies a little bit on what you do in a hierarchy. But, this time, rather than the parent, you use the old key. The old key introduces the new key by signing across it and you publish the new key

in the root zone file and do so for a minimum of 30 days. We actually did it for 395 days, just to make sure. So, it's been there for a while.

After you've managed to do this long enough, you are able to withdraw the old key, use the new key, and we're all done. Should be great, right? So should this clicker. Too far.

So, like many questions about the DNS, it's not easy to find answers. The problem in attempting to roll this key, as ICANN and many other folk have been interested in, is the key question of whether there are users out there whose DNS will go black. Just nothing will work anymore. Because if all of your recursive resolvers perform DNSSEC validation and none of them have learned the new key, when your cache is expired after October 11, there was no DNS for you.

Now, that's after the event. But what ICANN wanted to know and what we all wanted to know was, before it happened, could we measure what was likely to happen as damage? That hypothetical question. And it's not a question about resolvers. It's a question about users. Because in the DNS, users and resolvers are different. Next slide. Because I'm going to give up on this. I clicked again.

So, what we would like the DNS to be. DNS 101. You're the client. There's a resolver. There's a server. You ask the resolver a question. The resolver asks the server. Back comes an answer. It's all wonderful. This is, of course, complete nonsense because it never, ever, ever works that way.

We are really good at complicating technology and this is a tiny glimpse inside the wonderful infrastructure of the DNS and even then I'm sure I've not got everything right. Resolvers ask other resolvers. There are resolver farms. If you think Google's public DNS on 8.8.8.8 is one machine, think again. It's an entire battery of machines all over the world. Most large systems have now got load balances, forwarders, and the paths within them are extremely difficult to find out.

The DNS doesn't have a time to live. If you set up a forwarder loop, I forward to you, you forward to him, he forwards to me. Nothing will stop it. For years. Ever. There may well be queries that are 20 years old in the DNS. We just don't know. It just happens.

So, when you start thinking about how do you measure this mess, it becomes a really challenging problem. The DNS is not helping us.

There are really only two points that are visible in the DNS. One of them is authoritative name servers, the servers for those gTLDs and ccTLDs and the root servers. Once a year in the day in the life, the root servers publish the queries that they receive.

So, if we can send information towards the root servers, we can find out something about the DNS. So, if we can attach information to the query or make a special query, that's one way of peering into the DNS to see what's going on. But that's not the only way.

The other way is in answers. Instead of changing the question, you change the nature of the answer and send the result back to the user.

Now, I kind of like this myself, because quite frankly, sending answers to the root zone doesn't help you and it doesn't help me and it doesn't help anyone except if you're running the server that has the answers. Sending it back to the user seems a little bit more natural to me. You can measure. I can measure. Anyone can measure because the answers are coming back to you. You can run the diagnostic web page. It's up to you. And to my mind, that seems a far more robust way of doing it. But [there were] these two methods.

First one, RFC8145, defined a little bit over a year ago and it required resolvers to alter the way they behaved. If they understood what was going on, then periodically, they would send a very special query – started with an [underscore] character, so obviously it's very special – towards the root servers. Any root server will do. And the information will be collected at the root servers and by the root servers.

When we looked at that data in September of 2017, it was sufficient to set off some really strong alarm bells because after the end of 30 days, the folk who are trusting just the old key flipped over to trust the old key and the new key. The [red] headed back to zero. Old key only. But, it didn't go to zero. There was still this pool of resolvers that were reporting they hadn't learned the new key. They were reporting that when we rolled the key, this pool of resolvers, up to 8% of all resolvers around the world, weren't listening, weren't going to work, would die. And the users behind that would equally have a problem.

Based on that data and the lack I suppose of truly understanding what was going on, ICANN decided to defer the key roll. That's why it's taken 395 days rather than 365 days to get through this process.

This is a longer-term graph of looking at all of this up until a few days ago. The black line is the fascinating line. Even right up to the date of the key roll, this method of signaling was saying that 5% of resolvers would have died. If that was the only data we had, we probably wouldn't have even rolled now. But, as with anything in the DNS, measuring the DNS is really hard because there are a number of questions about that data. It's clear that a bunch of resolvers are saying don't know anything about that key. If you roll it, something is going to happen. But, is the resolver a validating resolver or is it just deciding to send some queries to the root?

Good question. In this wonderful world of allowing techies to be creative in software, what is ever is possible will be implemented by somebody sometime, anytime, and we will see the results. So, is it an accurate [signal] about the state of the resolver? No, not really. Is it an accurate [signal] about the identity of the resolver? Remember forwarders? No, not really. It's not very accurate.

And the really tough one, I have a resolver at home. It's me. One client. The company that Warren works for, Google, runs all eights. That has more than one client. It has around 14% of the world using it. So, while my resolver can quite happily be sacrificed, it's dead. I'm the only casualty. On those big ones, the ones with lots of users, you've really got to be worried because if they ever have a problem, the whole bunch

of users behind it will also have a problem. So, we really need to understand when we get resolver signals how many users are behind it.

The other thing, too. I don't know if you've looked lately, but when you get a resolver config and you look at how many resolvers did my ISP give me, how many were downloaded when I booted? What's in that config file? We call it [inaudible]. It would be a rare case of only having one. You might find two. If your ISP is feeling generous, you might find three, four, five. Programmers are endlessly inventive. I'm sure someone has ten. Ten doesn't help. Two is a good number.

But, the issue is even if one dies, the other one might be your salvation. So, even if one of my resolvers hasn't managed the key roll, as long as the other one has, I'm okay. You're okay. So, the big question here is not about resolvers. It's about users and the proportion of users.

Why is this hard? Because the DNS is an amazingly obscure protocol. You can't track queries through the DNS. They are untraceable. Because to make it fast, almost every single active element hangs on to an answer and replays it. If I ask a question once, it might take a bit of time. If I ask the same question, soon after, it will be lightning fast because the answer will have been remembered by all the resolvers in the path. The first one in the path [inaudible]. Caching changes everything. But, that suppresses signals back to the root.

The other thing about that kind of signal is that it measures resolvers, not user impact, so it's really not worth it. It doesn't give us data we need. So, can we measure users? Can we devise a signal that signals back? The answer, again, is no. The DNS doesn't do that for you. The

DNS doesn't give you a special signal based on your KSK trust point. We need to change the DNS.

Now, this change isn't an RFC even today. So, while we devised a mechanism where if you send a very, very special query, that it has exactly those characters, root-key-sentinel-is-ta, root-key-sentinel-not-ta, if you send those queries to a validating resolver, it will either answer or send back failure depending on whether or not it trusts that key locally.

Great idea. But we only really had the idea in July. The key roll scheduled for October and everyone had to change their resolver code in time. Sigh. So, it was all happening too quickly.

The other thing that really worried us is that it's only the new resolvers that will do this, but let's press on. How do we do this? We get you to run a script. Go to a special web page. What was the name of that web page, Warren? Is he there still? He's gone. Kskroll.info I think, but don't quote me. Find him and ask him. And you could run a web page that said your resolver system is going to survive, your resolver system is going to die, because you can test yourself.

But, there's another way of doing this and that's actually to look a little bit deeper in a script, and one way to do this is a script using ads. So, we deployed – you've got the URL. Thanks, Tim.

TIM APRIL: ksktest.net.

GEOFF HUSTON: ksktest.net. It's rolled. If that gives you the wrong answer, something is deeply broken in your DNS, okay? Getting back to this. We used ads. We did what you do on the web page inside the script in an online ad and we presented the ad around seven million times a day across the entire Internet, across a period since the 19th of September.

There's a number of fascinating facts about users. The first thing is the DNSSEC is remarkably well deployed, almost as good as v6 these days. 16% of all users use DNSSEC validating resolvers, and if it is not validly signed, they will not go there. Even more, actually, try and validate, but if they don't like the answer, they just find another resolver that doesn't validate, which is cheating.

16% of users won't go to an invalid and what we found is that half a percent of users were behind resolvers that had done an early adoption. [Knot], Unbound, and BIND had all done this feature in time, as I recall, and a few folk deployed it really quickly, .005% of users were saying they might have had a problem.

So, if we put that on a graph and extrapolate forward, what we were looking at wasn't 5%, wasn't 8%. It was a lot, lot lower. Between .1% and .2% of users were saying they might have a problem. But, as with the DNS, there's much more noise than signal. It was really unclear even in those measurements that that .1 to .2% was real or just an artifact of the DNS being the DNS. Very, very hard to tell. So, a lot of uncertainty.

The other thing about this, too, was that we were assuming that everyone was going to jump before the key rolled. But some of you like leaving things to the last second. I wish you wouldn't. Some of you were doing key loading manually and said, "I'm not going to do this until October 11th." Oh, geez, thanks. So, we were actually getting some signal, but folk were on top of it. It was their resolver that were doing this manually.

And the other thing, as I said before, we're actually looking for those resolvers in your dusty cupboards, resolvers that haven't been touched for years. The problem is that this is new code and you haven't loaded the new code to the old resolvers, so we weren't actually looking for the things that we needed to see. We were looking at folk who were on the ball, who keep their software up to date.

So, October the 11th came and went. This is a graph from SIDN. This shows the DNS key queries and it shows the signatures changing over. The roll happened across a cache expiration period of 48 hours. [Life] went fine.

What did I see? I saw .01% of folk after the roll were still saying they had a problem. This, of course, is nonsense. It's really noise. So, that tiny amount of signal that I was seeing was actually noise signal coming out of the experiment itself, which again, just goes to show the DNS is endlessly fascinating and trying to get reliable measurements when your tolerance is less than one in 1,000, one in 10,000 to one in 100,000. Most measurement systems are incapable of running at the at degree of accuracy when you're looking from the user side.

What did we learn? We panicked. Bringing up new software in the DNS two months before a key roll is kind of heroic and stupid. We were forced to do this because we needed better data to give folks confidence. We were kind of cornered. It was a bad place to be. Don't do this again.

Measuring resolvers isn't the right answer when you want to look at the way DNS works for users, measure users. And if you're feeling cocky and you think we should roll this every few months, just remember we were lucky. Just remember, to some extent, there was an extraordinary amount of effort to inform folk.

We've done one thing. We really have flushed out folk who have statically written the old key into their configs. Those resolvers, even if power is being applied to them, are dead resolvers if they're validating. They're gone.

But, instilling a habit of either doing automatic tracking of keys, or if you feel like doing it, manually changing the keys for every roll is still going to be an effort.

So, I suggest we do roll and I suggest we roll every year. It seems like a reasonable operating practice. But, just remember, we can't just let it happen automatically. It will require a little bit more care and attention.

So, my idea, keep it rolling. Just keep on tracking it. But, remember, this is not just going to happen automatically the second time or the third. It will take a bit more effort. Thank you.

EBERHARD LISSE: Thank you very much. I like the idea of rolling it every year because if it becomes like an RFC type of policy, everybody gets used to it.

GEOFF HUSTON: That is certainly the case. There are two reasons why we want to roll. One is no key is eternal, but the other thing is there is the ever so slight chance of having the existing keys unavailable when you need them, or they're being compromised or some other form of totally unanticipated event. Having a discipline of understanding how to change the keys is better than having none. How it will be useful? I don't know. But, it's better than having nothing in cupboard.

WES HARDAKER: Wes Hardaker, ISI. I love your work, as always, Geoff. You have fascinating numbers and fascinating graphs that I could stare at all day long.

We do have to be careful, though, because there is no … And you said it, but I'm going to harp on it again. There is no good measurement system that will cover everything. The nice thing about your system is that you cover most users, but not all of them, and that's okay. You know that caveat.

But, we don't know how many other systems out there use the DNS in infrastructure, in storing serial numbers, and they use it as a database, that they could have gone offline and there's no way of measuring whether they did or not.

So, here's the hypothetical one that I just came up with. During the time that a resolver that was only serving mail actually had only the old key, all of their reverse spam lookups would have suddenly not returned, the server was bad, and they would have accepted spam that they would not have. So, there's a bunch of things that we simply can't measure unless we actually bind things together.

So, all of the lookups right now, unless we actually manage to somehow transfer from the resolver with a key, I'm looking up the key. This is how I'm going to validate and these are all my trust anchors. Unless we bind all of those into one packet, we can never measure it perfectly.

GEOFF HUSTON: I agree, Wes, and certainly I don't think any of us are capable of measuring the Internet anymore, and if you include things, all kinds of automated systems and make use of the DNS, any technique is going to only reap some kind of subset. And yes, there is a certain amount of what we call lamp post bias. If you're looking for something that's lost at night, look under the lamp post because that's where the light is. It's not necessarily where you lost it, but that's where the light is. We always have this problem in the DNS.

However, to think that we could ever measure the DNS to any degree of accuracy is, I suspect, going a little bit too far. What we were trying to do in this exercise was trying to give the decision-makers, and ultimately the ICANN board, some sense of the degree of risk. That's all we were trying to do. We were not trying to say that resolver, that resolver, and that user have a problem. That's too far to go. But, we

were trying to understand from the bits of the Internet that we were capable of seeing to what extent we saw there might have been some risk.

And, yes, Wes, that's about as best as we can do in the DNS. It's frustrating, but that's where we are. Thank you.

EBERHARD LISSE: If you want to [inaudible] the last question, the more measurement you repeat, the more you can refine your understanding of accuracy of a measurement, isn't it?

GEOFF HUSTON: To some extent, the more you repeat, the more that's true, but when you try and script a user, let me tell you, particularly in ads, you guys have the attention span of a very small fly. And if I want to do something on your web page for more than a couple of seconds, you're gone. You're out of it. You're looking at some other YouTube thing or whatever.

So, repeatable measurements, scripting users, are incredibly different. If I create a web page that you all go to, only the geeks go there and I get a bias towards folk who really understand the DNS and want to measure themselves. I don't get everybody.

So, all kinds of measurements have this innate bias. Repeatable measurements are extremely hard. What we have tried to do in the ad system is compensate by the law of extremely large numbers. Measure

as many people every day as humanly possible to at least say that there is some wisdom in crowds, if nothing else. Thank you.

EBERHARD LISSE: I'm taking one more question from Jaap.

JAAP AKKERHUIS: Jaap Akkerhuis, [inaudible]. Just from KSK [inaudible] dot-net here locally is very interesting. It doesn't work because it cannot find which of the two keys it actually should use. Just something very [inaudible] resolve we are all using happening at ICANN.

GEOFF HUSTON: Brokenness is always with us.

UNIDENTIFIED MALE: Very quick.

EBERHARD LISSE: No. I said this was the last question. We are running out of time. Thank you very much, Geoff. You can take it offline with Geoff anyway. Hilde Thunem is going to make your last presentation.

HILDE THUNEM: Okay. Thank you for being here and listening to the last presentation of the day. I'm going to talk about the sexy, sexy issue of GDPR and WHOIS, the really hot topic, obviously, in everybody's mind.

What I will talk about is the [inaudible]. I am the managing director of dot-NO, so we have chosen our approach to WHOIS in a post-GDPR world. This approach may not fit for everyone, but there might be some other things that we have done that could be helpful for other TLDs.

Now, like any registry, we process customer data and we do this mainly in order for people to be able to register domain names and also because we want to manage the top-level domain in a way that contributes to the robust operation of Internet. So, those two are main reasons for us to actually have customer data.

Before I start talking about what we show through a registration data directory service, we need to have a little look about what data we actually collect and what that tells us about the domain holders.

So, at the core, we have the data on the domain registration. This, I think, is what everybody has. We do know the domain name that's registered. We have DNSSEC information for people that have signed their zones. Registration date, last update, etc. So, this is the information on the registration itself.

Then, we have a rule that the domain holder can be an organization or an individual. We have a presence requirement. So, we identified a holder through a unique identifier. Either it is registered in the business registry, and foreign companies can do that as well, or it is a person with a national identity number in the national identity register. We look that up in the register and then we give them a unique identifier that they can use towards the registrar so as not to expose their national ID. But, at the bottom, we do know who the holder is. Mickey Mouse does not

have a dot-NO domain name. Then, we have contact information on the holder.

We have had, I would say, a pretty standard data model. As most registries, we called our admin contact for a legal contact, just to confuse everybody. But, it was the same type of principle. We had a domain holder. We had a legal contact that was a person. We had a technical contact that could be a person or a role. And if you registered a role, like a domain host master, you could add lots of persons behind that role showing us that currently this role is [inaudible] specific persons and we have registrars.

We took a look at that and thought what data do we actually need to do our job? We decided that we do not really need the legal contact. We want to have a domain holder. If it's an individual, we have a name. If it is an organization, we want to have the name of the organization, the organization unique identifier, and the name of a contact person inside that organization. But we do not need whole separate set of contact data for that person.

So, we cleaned up that. We said that the technical contact, that's really the contact point where there's technical issues with the domain name. This does not have to be a person. We don't really actually want it to be a person. We want it to be a role. We don't want to know who exactly in your company is running the domain host master or whatever, so just give us the role information. Then, the registrars and name servers.

Of course, going from that model to the new model took a little bit of clean-up. So, we have deleted roughly 550,000 [inaudible] from our

database. We started out with 330,000 domain names that had only technical contacts that were persons and a lot that had roles. But we have fantastic registrars. Currently, we have 12,000 domain names that have persons as tech contacts and we are in the last cleaning of that.

So, what do we actually show as a registration data directory service? Well, why do we have it? This is what a lot of people ask themselves. We have it because we want to contribute to resolving technical problems, so if a domain name is doing things, attacking other domains, interfering with the way the Internet works, we want there to be a publicly available point of contact, that you don't have to call us in the middle of the night to get. So, something you could actually look up and use.

Then, we also think that if you have a domain name, you have the exclusive right of use to a unique piece of the Internet. Small piece, but still, so you can definitely want not people to know who you are, but you do not have the right to be non-contactable. So, we also say that in order to have a dot-NO domain name, somebody in the public may contact you, whether they are law enforcement, authorities, just somebody that wants to buy your domain name, just somebody that wants to send you an e-mail. This is a part of what you have to do if you have a domain name. So, this is why we have a service.

This is the information we show. So, like an overview. We showed basic set of information about the registration. That's the blue box. Then, depending on who the domain name holder is, we show various amounts of information about the holder. So, if it's one of the 63% of

the domains in the zone that's held by an organization, a legal person, not a person at all, then we show everything except the name of the contact person. If it's the 27% of domain names that is held by a sole proprietorship, that's a single person doing business, then we will only show the organization name, the number, the e-mail address, and the country. We have a presence requirement, so the country is Norway. And if it's an individual, then we will show an e-mail address that you can use for contact and country, but no name, nothing else. And an e-mail can of course be not very identifying if the person that has registered the domain name doesn't want to identify themselves.

Then, for all types of registration, we showed the name servers, the technical contact. Not the postal address of the technical contact. We don't think people usually need that. But the country so they know something about the time zone. And the registrar.

This is the information that is available but not all information is available through every channel because we are using two different technical channels, a standard WHOIS at Port 43 and a web interface, and we're trying to tailor those channels accordingly to what target do they have. What is target group of the channel and what potential there is for misuse.

So, this is standard WHOIS. It's intended for the technical community. So, it's written or it's following the standard way of communicating in WHOIS, which is techie readable and not human readable, although techies are humans. And the goal of the service is to contribute to resolving technical problems. So, this means that every lookup gives

only the information requested. You put in a domain name. You get a small blood of information, giving you handles for the name servers, for the registrars, etc. So, if you want to know who the registrar is, you then have to take the registrar handle and make another request. It's fairly stupid. It just gives you exactly what you asked for.

It's possible to do automatic lookups, and this of course creates a potential for misuse because you cannot use something like CAPTCHA on a service like this. And limiting the amount of requests that a single IP address can make is only of somewhat limited effect in reducing abuse So, that's why we show no information about the domain holder at all. You don't get any information about who is the registrant of the domain by using this channel. That you have to go to the web interface, and the web interface, of course, can use the strength of a web interface which is that it can actually try to be more public, human readable. So, it is mainly there for providing the opportunity to contact the domain holder, and if it is an organization, see who's behind the domain. It can also be used, of course, to resolve technical problems.

This time, we take all the information connected to the domain and we [inaudible] it together and we show everything for a single lookup because the normal users are impatient and they don't want to do the [serial lookups].

But, we do [emphasize] the things we think they need, which is who is the holder of domain and who's the registrar. Then, you can click and get a sort of [inaudible] screen showing the technical contacts and the other information.

This is also where we use CAPTCHA, we use fairly strict rate limits, because we think when you're using this service, you're probably just looking for a few domain names, mostly. Or if you're a legitimate user, at least.

If we have an individual, then you get much less information. So, I don't know how easy it is to read, but basically we say about the holder that – the holder is a private individual, so you will get the country, which is Norway, and you will get the e-mail address. In this case, the e-mail address is delivered by a provider that puts on anonymized e-mail addresses. And that's a service that many of the registrars offer, for example. So, this is somewhere where you cannot see who is the person behind the registration, but you can still contact them.

Of course, there are also private individuals that have e-mail addresses, more like hilda.thunem@something and that's also perfectly okay. It's their choice.

Then, we can use the web interface to offer extra functionality so you can put in the organization number of a domain holder. You get [inaudible] domain names per registrar. This is a hospital, which is why they're using lots of different registrars, because they don't have an internal IT department that manages to sort of streamline it. And you can flick the little button to see which of the domains are signed. So, that's the one with the green mark.

So, [inaudible] access. The wave of the future. RDAP, solution to everything, at least is what is been [inaudible] us. I think it's [inaudible] but it's just a technology.

We are considered layered access, and in a way, sort of, we already have one because we do have a special WHOIS service set up for the registrars that show them a lot more information. Then, we, as a registry, use that as well. Then, the whois.norway.no is another instance of WHOIS showing just small amount of thing. And the web interface runs on the registrar WHOIS with something else. So, it's not quite as [inaudible] as we would like. And we think that RDAP, in a way, could be a way forward with that. But, basically, layered access is about showing different amounts of information to different groups. That's what we already do.

So, before we start going heavily into RDAP, we will be doing launching an RDAP pilot at the end of October, but going more into actually building layers. We need to have more of a discussion inside of Norway of are there other layers we need to build. That's where you get into are there other groups that have legitimate interest to have access all the time, which is different than having legitimate interest to just get the answer to a single question about a domain name and how does this scale.

Then, of course, we will be looking at RDAP as a technology to replace WHOIS sometime in the future and saying that, well, if we can do web interface, we'll run happily on RDAP as well, if we make an RDAP service that shows the same type of information that whois.norway.no does today and shows more information to the registrars through authentication. Do we really need to have a WHOIS service running on Port 43 forever or can we turn off the lights and close it down sometime in the future?

So, that's it. That's more information if you want to read about the way we have balanced privacy versus openness and the way the web interface run. It's there. It's the terms and conditions.

EBERHARD LISSE: Thank you very much. Any questions from the floor? I understand this e-mail issue, what e-mail [inaudible] to use has been [inaudible] in a recommendation or in a letter by the data commissioner's group to ICANN. They can say, they have said e-mail you [can] publish because it's the choice of the registrant what e-mail [inaudible] to use, if you want to use private or generic one. [inaudible] can do so. So, I think that's very helpful.

HILDE THUNEM: Yeah. That's what we got from our data protection authority because we had a discussion with the Norwegian Data Protection Authority when we made this service on what to show, what not to show, and we did discuss e-mail with them, and they agreed that as long as the market provides solutions for anonymized e-mails. And it does because registrars do it. You can register 12345678@gmail.com although as a ccTLD manager, I sort of die a little bit every time there's a Gmail address. But, still, you can register that. You can do info@ at your domain. There are so many ways that you, as a user, can analyze your e-mail that we, as a registry, did not have to provide that. And we were happy for that because we did not really want to be in a situation where e-mails are sent to us that we should forward, because then we're suddenly processing more personal information that we don't really

feel we have a reason to do. But I know that here is where other registries have chosen different solutions.

EBERHARD LISSE: Okie-doke. Thank you very much. [inaudible], you can wrap up. [off mic].

UNIDENTIFIED MALE: Yeah. Well, I did make some notes, actually.

EBERAHRD LISSE: Even better.

UNIDENTIFIED MALE: Yes. [inaudible] for doing this for years and finally we can [inaudible], so congratulations.

EBERHARD LISSE: Thank you.

UNIDENTIFIED MALE: And congratulations to [inaudible] and the rest of the program committee for putting together such a great program. Thank you.

I'm [inaudible] presentations so I have to go very quick through them. But even better, we had 134 visitors this morning, a little bit less today. I think that's a record.

EBERHARD LISSE: I don't know, but 134 with standing room only is perfectly in order and I'm going to make it a point to come tomorrow when the ccNSO is sitting in the same room.

UNIDENTIFIED MALE: So, after listening today and trying to find a general trend, I find basically three key words that are not just important in this room, but also in the rest of the Internet and the rest of the society out there and that's based on security, trust, and risk. I think that goes through all 14 presentations.

So, the first presentation on a bug bounty program or maybe a responsible disclosure policy. You don't even have to pay people money. It's all about keeping your own systems safe.

And we had a presentation from dotCAT about how to keep their users safe by WHOIS privacy. Of course, you have to deliver the right data to the right users and avoid zone poisoning. You have to keep your servers running. So, we had a presentation about Anycast sinkholing. It's always fun to watch Jacques presentation about home gateway and keeping your host safe. And I really have to talk to him about getting a [inaudible]. That looks very interesting.

Next generation firewalls and open source, software for doing those. A very interesting talk there. The most important part today, lunch, and the sponsor, SerNet. Thank you very much for sponsoring lunch. [applause]

During the AU transition talk, the only word I kept hearing was testing, testing, testing at all levels, at all points, for very many months. That was very interesting to hear. We had the host presentation from dot-ES. Very interesting to see the hybrid solution they took to the cloud. We've seen other registries thinking for moving to the cloud but always leaving one provider and using basically infrastructure as a service or sometimes a [bit hybrid] but only using one provider, so very interesting to see a [hybrid] solution there.

An update on RDAP. It looks like it's finally happening. I think we're all very happy to see the [inaudible] protocol of WHOIS disappearing.

We're up to number ten now, Nominet. Also very interesting to see a very easy way to get started on running a global Anycast service yourself without actually putting hardware at many places in the world and keeping those running. An open design of the hardware part of the HSM. It's also finally looking like a finalized product. It's also a very good update on that project.

IDN homographs. We've seen a few attacks before. Now it looks like there's thousands out there, so it's a real problem we need to start fixing. And for those who were not in Puerto Rico, go back to Tech Day in Puerto Rico and find a talk by Patrik Falstrom about emoji, which is even more interesting and sometimes [inaudible] of what [inaudible] create domain names.

The KSK rollover happened and I'm always very happy to hear Geoff talk and nice graphs. I keep being amazed how this thing actually keeps working.

Finally, presentation by NORID. It goes to one of the good things about GDPR and that is to go back and think about what data you collect and why do you actually need it because it goes back to the very essential thing about security. Less is more.

And talking about security, I'm pretty sure we're not finished and I'm pretty sure we'll see [inaudible] again and talk about security in Kobe.

EBERHARD LISSE: Thank you very much. That's it for today.

**[END OF TRANSCRIPTION]**