BARCELONA – DNSSEC for Everybody: A Beginner's Guide
Sunday, October 21, 2018 – 15:15 to 16:45 CEST
ICANN63 | Barcelona, Spain

WES HARDAKER: … Makes the whole world secure if you use it and turn it on. We're going to do that through a story. We're going to start with the origins of DNSSEC, which started in 5,000 B.C., quite a while ago. Before the Internet even, DNSSEC already had its beginnings.

We're going to introduce Ugwina. She lives on the edge of the Grand Canyon. For those that aren't familiar with the Grand Canyon, it's a gigantic whole in the middle of the United States.

This is Og. He lives in a cave on the other side of the Grand Canyon. It's a long way down and a long way around, so they don't get together much to talk. I've done it. It takes about two or three days to climb from one side to the other.

On one of their rare visits, they notice the smoke coming from Og's fire. Soon, they are chattering regularly using smoke signals over UDP. Until, one day, mischievous caveman Kaminsky moves in next door to Ug and starts sending smoke signals, too. For those that weren't around ten years ago or whenever, Kaminsky was the person that really discovered a pretty significant vulnerability in DNS.

Now, Ugwina is really confused. She doesn't know which smoke signal to believe. Which one, from really far away, without binoculars to identify who's sending it, should she read?

So, Ugwina sets off down the canyon to try and sort out the whole mess. Ugwina and Og consult the wise village elders. Caveman Diffie thinks he might have a cunning idea. For those who don't know, Diffie was one of the people that created public key encryption, which is the foundation of much of cryptography within the Internet today.

In a flash, caveman Diffie jumps up and runs into Ug's cave. Right in the back, he finds a pile of strangely colored sand that has never been found in Ug's cave. With a skip, he rushes out and throw some of the magic sand on to the fire, and it turns into a magnificent blue smoke.

Now Ugwina and Og can chat happily again, safe in the knowledge that nobody can interfere with their conversations, because only he has the magic blue sand.

So, that's the introduction to DNSSEC in silly diagram form. So, let's go back to DNS a little bit because it actually works in sort of a similar way as smoke signals.

There's sort of a tree structure in the DNS, as some people like to call it, or a hierarchy. It starts with the root at the top. Then, underneath that are all the TLDs. That includes the ccTLDs, the gTLDs, and the new TLDs. Then, underneath each of those are registration points where various things happen.

So, I'm going to point you to the middle on in the bottom, which is bigbank.com. We're going to come back to that a few times. So, the root delegates com information to the com servers, and the com servers delegate bigbank to their servers.

A resolver, when it's trying to answer your request for maybe where www.bigbank.com is, it doesn't know the whole tree. It has to start from the top. But, it knows where the root zone is, and it's able to traverse the DNS hierarchy in order to find it.

So, each level in that hierarchy, as I said, refers to the next resolver. So, the resolver can simply follow the chain all the way down until the question has finally been answered.

The resolver caches all that information for future use. So, once it actually has that knowledge, it can answer you much more quickly for a while because it will cache the information for as long as it is able.

There's a problem with the DNS, though. It was not designed with any form a security. Just like the smoke signals had no security in them, the DNS doesn't either. Naively, it just assumes that everything is correct. Names are easily spoofed, and those caches that remember stuff for a long time, if it gets a bad answer in the cache, it's going to remember that bad answer for a long time, too.

So, because the silly diagrams weren't silly enough, we're actually going to do a skit. This is where it gets really fun. How many people have seen this skit before?

A couple of you. Good. The rest of you in are in for a treat. The few of you that have seen it before are going to have to suffer and watch it again.

All right. Could I get my volunteer players to please stand up?

So, we have a few players. We have an average user, Joe User. Please raise your hand. He is going to be the user in this scenario, and he needs to do some banking today. We have a root representative that represents the root of the DNS tree. We have an ISP that knows where the root is. We have dot-com, and we have bigbank.com. Actually, you're probably best right there, Fred. So, I'm going to let them take it over at this point and kick it off. Go for it.

UNIDENTIFIED MALE: [When] the light comes one, hello, hello.

WES HARDAKER: Oh, there we go. Okay, they're both working. Here, Tim. Russ has the queries.

TIM: Wow! I have all this money. I need to make a deposit. I need to go to www.bigbank.com to deposit my money.

UNIDENTIFIED MALE: Well, I don't know where that is, so I'll go find out for you. So, I'll start with the root. Hi, do you know where www.bigbank.com is?

FRED BAKER: Wow. I have no idea. However, have I got a deal for you! You could ask dot-com, and dot-com might know. He's at 1.1.1.

TIM: Perfect. I'll give them a try.

WES HARDAKER: [inaudible]. There you go. Sorry.

TIM: I've been told to ask you. Do you know where www.bigbank.com is?

UNIDENTIFIED MALE: That's a great question. I'm dot-com. I'm at 1.1.1. Thank you for visiting me. I can tell you where bigbank is. Here you go. Bigbank is 2.2.2. Wish to go there?

TIM: Thanks. I need to know where www.bigbank.com is.

RUSS MUNDY: Well, thank you for asking, Mr. ISP. As a matter of fact, I can tell you exactly where www.bigbank.com is. It is 2.2.2.3.

TIM8: Thanks. Bigbank.com is at 2.2.2.3. Outstanding! I can go ahead and deposit my money! Thank you, Bigbank!

FRED BAKER: I don't know. He should thank you.

WES HARDAKER: All right. Thank you, peoples. Stay around because we're going to need you again in a minute.

UNIDENTIFIED MALE: Don't quit your day job!

WES HARDAKER: This is their day job. So, Ugwina, the resolver, when chatting with Og, the server, is sort of similar to what you just watched. They were communicating over smoke signals. There was no problem. There was nobody interfering.

Next, we're going to see what happens when, maybe, something goes wrong. Can you guys please repeat your performance?

TIM: Wow! I now I have one million dollars, and I need to deposit this money. I need to go to www.bigbank.com, so I'm going to go to my ISP and ask, "How do I get to www.bigbank.com?"

UNIDENTIFIED MALE: I have no idea, but I'll try. Hi. Do you know where www.bigbank.com is?

FRED BAKER: Well, I am the root, and I really don't know such things. But, you might ask dot-com. He might know. He's at 1.1.1.1.

TIM: Okay. Thank you. Hi, dot-com. Do you know where www.bigbank.com is?

UNIDENTIFIED MALE: Welcome. I'm dot-com. I'll tell you where bigbank.com is. Bigbank.com is at 2.2.2.

UNIDENTIFIED MALE: .2.

UNIDENTIFIED MALE: .2.

UNIDENTIFIED MALE: IPv3?

TIM: Thanks. Hi

UNIDENTIFIED MALE: Hi.

TIM: Hi. Do you know where www.bigbank.com is?

UNIDENTIFIED MALE: I sure do. It's at 6.6.6.6.

TIM: Perfect. Thank you. [inaudible]

UNIDENTIFIED MALE: There you go. www.bigbank.com is at 6.6.6.6.

UNIDENTIFIED MALE: Oh my gosh! Thank you! Going to deposit my money!

UNIDENTIFIED MALE: Thank you.

WES HARDAKER: All right. Thank you, again. So, the real question is, how are we going to get out of this horrible session with Dr. Evil presiding over every bank.

So, Ugwina the resolver is confused. This is a situation you just saw. There's two signals. She doesn't know which one to believe. Actually, in the DNS, whichever one returns first is usually the one you believe, or at least it was until DNSSEC came around.

So, again, going back to the same chart, if two different servers are giving you two different answers, you might get bigbank.com answer from two different places, and you don't know which one to believe. So, in the diagram above, there's a blue one and a red one. You just have to guess which answer is the correct one.

But, DNSSEC actually adds security to the DNS, and it uses digital signatures to do this. It basically ensures you two things, that information has not been tampered with, and that it originated from the right place. So, no matter how many places it's been stored, no matter caches it has been in, it has not been tampered with and you know where it came from.

The keys and signatures are themselves stored in the DNS as well. Since the DNS has a lookup system, the keys can be simply looked up, as well as the signatures, just like any other day, as long as you have a place to start.

So, the high-level concept of DNSSEC is that the resolver knows not just where the root server is but also what the root server's key is. As long as it knows that, those two bits of starting information, it can verify the rest of the tree.

It does this by building a chain of trust. Every level signs the next and the data involved in its own level, until the chain is complete all the way down to the answer that you need.

In that way, finally, the resolver at the ISP can actually determine which of those bottom-two bigbank records are correct, the correct one being the blue one, and the X indicating the bad one.

So, let's see what actually happens with DNSSEC in play. There's a couple of things that we're going to go on today. You can see that now we have these little medals. These indicate the digital signatures. Each of one of the three DNS servers is going to have this medal, and then the

ISP is going to have a matching medal to make sure that he can verify that he's talking to the right person. Take it away.

TIM: Wow! I have another deposit to make. I need to go to www.bigbank.com to make this deposit.

UNIDENTIFIED MALE: Okay. I'll look that up for you.

Hi, root. Can you tell me where www.bigbank.com is?

FRED BAKER: Well, okay. I'm the root, and I can prove the authenticity of my statements. But, I have no idea where bigbank is. You might ask dot-com. He's at 1.1.1.1.

TIM: Thank you. Hi, dot-com. Can you tell me where www.bigbank.com is?

UNIDENTIFIED MALE: Hi, ISP. I'm dot-com. Check that I'm dot-com. Bingo! I am! Great. Let me just tell you where bigbank is. Bigbank is at 2.2.2.

UNIDENTIFIED MALE: .2.

UNIDENTIFIED MALE: .2.

UNIDENTIFIED MALE: Hi, bigbank. Can you tell me where www.bigbank.com is?

UNIDENTIFIED MALE: Sure. But, I have no sig, so you should reject me.

UNIDENTIFIED MALE: Oh, no! Goodbye. Hi, bigbank. Can you tell me where www.bigbank.com is?

RUSS MUNDY: As a matter of fact, www.bigbank.com is at 2.2.2.3.

TIM: Thanks. Can I verify your signature?

UNIDENTIFIED MALE: Ding!

UNIDENTIFIED MALE: www.bigbank.com is at 2.2.2.3, and it's signed.

TIM: Excellent! I can go ahead and deposit my money safely.

UNIDENTIFIED MALE: Moneeyyyyyy!

WES HARDAKER: All right. Please give a hand to the final act of our three-part play. I'll take knowledge that we need the little dinger in order to verify the things when they clink together in the future. That would be great.

All right. Well, silly as that is, that's really what happens in the DNS system and really what happens with DNSSEC. Of course, it's a little bit more complex than that. But, in reality, the resolvers really do chain all the way down from the root. They go through whatever TLD is next and through multiple levels to get the final answer.

But, sometimes those levels can get quite deep, but as long as DNSSEC protects the chain all the way down, you end up getting a secure answer, just like the blue smoke protects the resolver from verifying that Og actually sent the real message.

So, at this point, I'm going to turn it over to Russ Mundy. I should have introduced myself first. So, I'm Wes Hardaker from the University of Southern California's Information Sciences Institute. Russ Mundy is from Parsons, and he's going to explain more about why you need DNSSEC.

RUSS MUNDY: Thanks, Wes. Also, thanks very much to our other players that were with us today. It was a lot of fun to do it with a new set of people because

this was a fresh cast. Thank you, everybody that joined us. I appreciate it, again.

So, this is the aspect of the DNSSEC for Everybody session that is intended to help people think about the reasons behind why anyone, whether it's an end user or an ISP or authoritative server operator would want to do DNSSEC validation, beside Dr. Evil jumping in the middle, because that's fundamentally what DNSSEC is designed to counter.

So, when you think about it, when we were talking about this in DNS terms, DNS doesn't move money back and forth between a user and a bank. There's applications that actually are used to do that.

So, when people are attacking DNS and substituting information, in almost all cases they are trying to do that because of other applications, other functionalities, that are done across the Internet.

So, if your DNS is not functioning properly, then, whether it's a web browser or an e-mail server or a chat engine of some sort, they won't work properly either. You won't get to the places where you want to get to or expect to get to.

So, that's the fundamental issue with why you want to get the DNS right: it's what needs to be right, so the rest of the applications that make use of it can also function correctly.

So, the hijack that our little skit is intended to demonstrate is really focused at getting these application software entities to talk to some place that the user at either end was not intending for them to talk to.

Over time, there's been a number of applications that have been attacked, with the attack starting at DNS. In fact, some of the attacks that start with DNS end up making exploitive type of use of two to three and sometimes four other protocols. But, it starts with DNS and getting the DNS information that the perpetrators want to get some place out there, where they can take advantage of the people they're trying to take advantage of.

So, at one time – I think it has gone from the Internet; I've not found it in the last couple of years – there was specifically a university course – in fact, there were two of them at two different universities – that required their CS students to actually write a DNS attack tool. That gives an example of relatively how easy is to do it. There is software that's available on the Internet that is DNS attack software.

But, the problem with the university classes, at least their online information that they had, is that there was nothing in there about the ethics aspect and how this was not something that you really should be doing, even the professor required them to produce this as part of their coursework.

So, it's unclear how much hijack software exists, but it does, and it's easily accessible or writable, for people who are interested in creating software.

So, when users are making use of DNSSEC, as Wes said earlier and has illustrated in the skit, the idea is having cryptographic information within the regular structure of DNS, because DNSSEC is a part of DNS. It is completely integrated with the standard DNS.

So, when the receiver of an answer gets a DNSSEC-signed answer, whether the validation takes place at a local validating resolver, or actually right in the application itself, the basic idea is that the user who sent the query for the information can gain cryptographic assurance that it came from where it was supposed to and that it hasn't been changed en route.

So, what this really does is this lets you be certain that, in fact, the answers that you got is the information that you should be using, whether you're doing a web query or whether it's mail servers talking to one another, or if it's a Jabber or a Twitter session or something like that.

So, as a still simplified example of how DNS exchange works, this shows Joe User doing his query. I didn't change the numbers to match what our T-shirts say. Sorry about that. Anyway, Joe User sends a query to his ISP, the recursive name server. The recursive name server then sends a query to the authoritative name server for the place that Joe User wants to get to. He gets an answer back from the authoritative name server to the name server and then returns that answer to Joe User. This was really the first scenario that you saw with the skit.

The next one after – oh, sorry. After the answers come back is when the data actually flows between the applications themselves. So, Joe User then talks to the web server and does his activity there.

Now, when you have a DNSSEC-signed set of zones and validating resolvers in play, you can tailor websites to give indications that, in fact, the correct information is being exchanged. By far, the vast majority of

websites that may be using DNSSEC do not have an indicator like this, but this was something that we did several years ago, just so people could readily see that they were getting a DNSSEC chain as they went back and forth. If you didn't have a DNSSEC chain underneath of it, you did a different indication when it came back.

So, part of the tailoring of the website was the insertion of an attack indicator. That attack is the same as Dr. Evil, who watches when a query is sent. A recursive server asks the authoritative server, but before the authoritative server even gets that query, Dr. Evil has given the answer to Joe User. So, Joe User goes to the wrong place. He goes to Dr. Evil's site.

In the meantime, the standard DNS queries are just happening out on the network, but Joe User's machine already has an answer, so it ignores the correct answer and sends the application to connect to Dr. Evil's site.

He never gets the information until DNSSEC comes into play. When DNSSEC comes into play, the Joe User rejects the bad answers you saw in the skit and then goes to the proper site.

There we go.

Now, in this case, the instrumented website has correct information in it, and, although we did it on the website, we instrumented it in a way that would illustrate going to a different website.

This is what a validating resolver would get, what you see with the green check. The first entry is, ".org shares Comcast DNSSEC advice for ISPs."

But, a non-validating resolver would see that Steve Crocker would say, "DNSSEC doesn't solve world hunger," as the first entry. You can see that the next one done the page is, ".org shares Comcast DNSSEC advice for ISPs."

So, essentially, information was inserted into the web page in a way that the user, without seeing the triangle at the top, would have absolutely no idea that their website had a DNS hijack attack, content substitution attack, raised against it.

So, you might say, "There's not a lot of DNS resolution, is there?" Well, this was about ten years ago when we measured CNN. This is about five years ago. For one page of content from CNN.org, it took that many DNS queries. So, there are a lot of queries.

The important information about the DNS and the DNSSEC – the critical piece – is that the zone data itself is what matters.

Another illustration that shows how the process works and where the changes need to occur. So, you can see, on the far left, that the zone information is put into the authoritative servers. It's then available on the DNS. So, recursive servers you see in the center talk to authoritative servers in response to questions that they receive from the client.

So, the client –as you can see, .1, the first arrow, number one. Number two, the recursive asks authoritative. Number three, the authoritative answers the recursive. And then, the recursive answers the client.

So, when you do your DNSSEC implementation, there's step that you go through that you add additional data and additional functionality. So,

if the implementation for your part of the DNS that you are either responsible for operating or responsible for the management of – it depends on where you are in the whole DNS structure of things what you need to do. Activities that are heavily and deeply engaged in DNS are probably going to want to do all of these actions themselves because they already have DNS-competent staff.

So, if the size and complexity of what you're doing is really tied tightly to DNS – you can see some examples there; [arch]-registry, big enterprise, activities with non-critical DNS zones that are easy to run and are small and easy to handle – they're the ones that are probably going to want to do a lot of this themselves, again, remembering that protecting the DNS zone data itself is the critical factor.

So, in the last illustration, where the zone information needs to be added to make this a DNSSEC-signed and validating function is – the signed information is put into the system by the owner of the authoritative server. It is then available to be included in responses. Then, the validating recursive server that you can see at the bottom is the one that in fact does the validation to say, "Yes. This is correct, cryptographically."

So, if the existing organization that you're looking at working with and dealing with has a lot of DNS activity and functionality and a lot of DNS centricity, then you can probably do DNSSEC internally, do it yourself.

If the activity has little or minimal use of DNS in terms of the core functionality, there's a good chance that there's an IT department, or that it's maybe outsourced in some manner. You should ask any of the

providers, whether it's your IT department or your outsource suppliers, if they are able to do DNSSEC. If they aren't, say, "You need to learn to do it, or I'm going to take my business elsewhere," because, in fact, there's enough people in the DNSSEC business now that that can become a very real possibility. You can choose to use suppliers that do DNSSEC.

That is the end of the main presentation part. We intend to have this very open question and answer session. We will – yeah. I'll turn this back over to Wes and let him be the main one pointing to people, because we have both a lot of not only DNSSEC players in the room but we also people who are very knowledgeable about DNSSEC.

So, please, ask any questions you have. We're fully ready to take on anything you might want to know.

WES HARDAKER: All right. So, DNSSEC is not yet – how about this one? Nope. Yeah. I need the microphones back on.

Test, test, test, test, test, test. Test – oh, there we go. Okay. All right. And does this one work? Good. So, we have two. I can use them both.

So, DNSSEC is not simple. There's a lot of complexity. We guarantee there's going to be questions in the room. So, we actually have a lot of time for the rest of this devoted toward just answering questions. We have a lot of experts in the room that helped created DNSSEC, so, pretty much, if you have a question, we can answer it.

So, anybody have questions? Raise your hand and we'll walk a microphone to you.

THANH NGUYEN: Hello. I'm Thanh Nguyen from Vietnam. Thank you for your presentation. It's very useful and there's a lot of information [inaudible] DNS.

I can see that a lot of [inaudible] of DNSSEC when deploying the [inaudible] to secure the ccTLDs and the gTLDs, but I think, when, in fact, you implement this in the DNSSEC, you need to think about [inaudible] [to some challengers]. It looks like you need to have a [inaudible] DNSSEC, DNS signing, zone signing, and some step I forgot.

But, based on this step, I think hackers or [inaudible] can attack by DDoS, or they can, based on the weakness of the DNSSEC, look [like] the time synchronization of DNSSEC because it takes time to put the – first of all, you put some additional points of the DNS system, but it takes time to wait for the response. So, a hacker can, based on this weakness, attack the gap in the DNSSEC system.

So, do you have any solution to solve this problem?

WES HARDAKER: So, I think you were asking about a couple of things. So, first off, you were asking about, is there a longer delay for users to request data, right? Because you have to request more data, it slows down the DNS lookup a little bit. It does, a little bit, not … the users that would care

the most are the ones that are using instantaneous applications, where they need to do a lot of DNS lookups and they need to look at all the responses and how fast. So, typically, that's web browsing and that type of stuff.

Remember that, because of caching, that's only going to slow down one lookup and that all of the rest of them beyond that, including the security records, will be cached.

So, the initial lookup may be on the order of milliseconds slower, but then it's cached beyond that. So, typically, if a delay is seen, it's only seen by one user at an ISP for one time, going to a website.

I think the other thing you were asking about is that some of the DNSSEC records are very large and that they can actually be used in amplification attack.

So, if you ask a question to a DNS server and it's returning a larger response, if you fake where you're coming from, then the response can go to the wrong person.

The typical solution for that these days is something called response rate limiting that most DNS servers today that won't let you query for so many records at once. So, if you try and query for 100 records all within a second, you're not doing something correct and I'm going to stop answering you.

So, that's typically the solution that has worked today. I'm actually operationally responsible for one of the root servers. The instant we turned that on, all of a sudden, we got a lot less requests and stuff like

that happening. And, that was years ago. So, that's almost a non-existent problem anymore.

Andrew, I think the –

ANDREW FRASER: Yeah. Did Warren want to add to that?

WARREN KUMARI: No.

ANDREW FRASER: No? Okay.

WES HARDAKER: Okay.

WARREN KUMARI: [inaudible] attacker send the response for [inaudible].

WES HARDAKER: You want to come sit up here, too? Any experts in the room are welcome to come sit at the table so that … go ahead.

EBERHARD BLOCHER: Hello. My name is Eberhard Blocher from Germany. I'm not an expert. I'm totally a layman.

WES HARDAKER: That's okay.

EBERHARD BLOCHER: [inaudible]. I have a question going back in history. I know DNSSEC has been around for a long time, I think ten years ago. Somebody told me Steve Crocker basically created it or was part of the people creating it.

Now, I understand, with a lot of time having gone by, that we now have a new situation. This year, all of my end customers I'm doing domain services for are asking for SSL encryption. So, now we have Let's Encrypt, which is free. So, basically, this is another way, this year, to secure websites.

I understand that the SSL certificate has to be linked to one domain name only, which means the question is, do we still need DNSSEC, or what is the difference between DNSSEC- and SSL-encrypted websites?

If I do encrypt my website using an SSL certificate, why do I still need DNSSEC?

The second question, perhaps – I've looked at statistics, and I know that many registries provide data for how many domains are DNSSEC-encrypted. There's some registries which have a very low penetration. So, why is that so? And, again, the same question: do we still need DNSSEC?

WES HARDAKER: Excellent question. So, one of the hardest things about Internet security as a whole is that you can't just solve one problem because it actually turns out that everything works in layers.

So, for example, if you do a DNS lookup – let's say you're going to an SSL-protected website. So, you know you got to the correct website, if you can get there. The thing is that DNS comes into play long before that, and it will actually redirect you to the wrong place entirely.

There's a number of issues with that. One, they can just send you to the wrong place and you'll never get to the right website. Routing has the same issue. So, if we don't protect the routing layer and the routing infrastructure – so, the reality is that the Internet is built on a stack. Really, every layer in that stack needs to be protected to fully protect end users.

So, yes, DNSSEC is still needed.

Do you want to talk to the history and the creation and maybe DNSSEC deployment and Steve Crocker? There's a lot of people went into DNSSEC. Steve definitely did help, but …

RUSS MUNDY: Well, yes. There is a long history of getting DNSSEC designed, implemented, and deployed. As Wes said, it's aimed at making certain that your DNS usage gets the proper answer to the user.

Now, what application is making use of it is relatively [independent.] So, it was actually in 1993 when the main focused efforts started on

getting DNSSEC developed and designed and laid out. There were calls made back and forth. Steve Crocker was one of the people that was involved in one of those calls. I happened to be one of the people, also.

But, there were multiple redesigns because this is the first ever to really insert an on-the-fly security capability into a running protocol. There were two designs that had some significant problems. The third one got it right.

In 2010 was when the root zone itself was signed. There were a couple TLDs signed before it.

So, yes, there's a lot of people that made many contributions. There's many organizations that have worked very hard at this. But, it is focused, we pointed, to prevent happening what's on this slide because, whether you're using an SSL certificate or not, there have been enough challenges with the certificate authorities, and erroneous or bad certificates issues, that you could have, in fact, this exact result occur and have a problematic CA involved in the website, and the user would still get an SSL-signed website.

WES HARDAKER: One other thing to note is that the web is not the only mechanism that needs to be protected for name lookups. E-mail is another good example that actually can't be solved with a traditional certificate system.

So, all be taking about that on Wednesday a little bit in the DNSSEC Workshop, which is something we'll talk about later, that everybody is welcome to attend. It's an all-day program, more about DNSSEC.

One of the things I'll be talking about how DNSSEC is really the only available solution to protect web servers from being with a man-in-the-middle and having your e-mail stolen and taken to the wrong place. The typical TLS certificates can't help there for a reason that I'll explain on that day. It's complex.

So, Andrew?

AHMAD ALSADEH: My name is Ahmad Alsadeh. I'm a Fellow. I'll ask why it is not widely deployed, like DNS. Why does not everybody deploy DNSSEC? This is the first question.

My other question is, do you believe that the new technology, like blockchain, could be a promising solution to secure DNS? Thank you.

WES HARDAKER: So, two parts to that. So, one, the other person asked for information about why some registries had a greater deployment and things like that. That's a really good question. Every region in the world seems to have a different rate of deployment and things like that.

Some TLDs – in particular, Sweden, the Czech Republic, and some others – actually gave you a financial incentive to sign your zone. So, they had much higher intakes and things like that.

A loft of it comes to, are there incentives? Is there are a local push to really push companies to use this technology? So, the adoption rate has been generally slow to push out.

But, we're actually at the point – again, if you come on Wednesday, I actually have numeric statistics, and I could actually pull up – well, it'd be hard to do right now. There's millions of zones that are now signed. So, even though the percentage is low because there are so many domains out there, we are into millions and millions of deployments. Like, 85% of the TLDs are signed, I think, at this point. So, it's getting there, but it's a long process to get the entire world to switch over.

RUSS MUNDY: One of the reasons, if I could add to that, is not just the TLDs and the second-level domains. The end users, the enterprises, have traditionally not been active demanders for DNSSEC because most of the time they just don't realize how important it is to their functionality.

But, in the last five years, the U.S. government – along the line what they did for creating the requirement to sign all .gov domains – and some of the other governments have taken a similar stand. But, the U.S. is where I'm most familiar with.

They also now require that validation be done for all of their domains. So, it's being pushed out to the end enterprises in some organizations. For instance, the company I work for, Parsons – parsons.com – is a signed domain and is operated that way.

So, it's keeping education going, encouraging people to ask for it. This is why you'll see several places say, "Ask your ISP or provider or whoever is doing your DNS service to provide you with DNSSEC capability."

WES HARDAKER: And, a lot of modern registrars and registries just have a button to push. If you're going to host your DNS with them, all you have to do is click the button and you'll have a signed zone. So, it's very easy.

But, for many people who run their own in-house DNS, there's a cost associated with deploying any security mechanism. It doesn't matter whether it's a TLS certificate or whether it's DNSSEC or whatever. You have to learn. You have to go about doing it. There's a lot of tools now that make it quite easy, but ten years there really wasn't. It was much more geeky at the time.

Any other questions? Yes?

UNIDENTIFIED MALE: Hola. My question is about the issue for comment about the last rollover of the key last week.

My other question is, is there any plan to add to the [browser a signature to the DNSSEC], to the domain?

WES HARDAKER: So, the key rollover happened last week, and it was the first one ever. It went quite smoothly. It turned out fairly well.

Do you want to talk to it, John?

JOHN LEVINE: I have the next question.

WES HARDAKER: Okay. You have the next question. So, the key rollover actually went quite well. There was a few impacted ISPs that actually hadn't change their key, but, typically, it's not the web browser that needs that key. It's the ISP, as you saw earlier in the day. Joe User, with a web browser, didn't really actually have to do much. He didn't even have to know that DNSSEC was happening.

And, there's a whole other issue there of, what happens of if you're in a wireless coffee shop and then your ISP is doing the DNSSEC, and are you protected there? That's a whole other program.

But, typically, only the ISP really needs to know about the new keys.

As far as ICANN's future process as to, are they going to roll it again, and at what date, that's going to be a discussion, I think, later this week, in fact. Over the next probably year, there will be a decision of, how often are we going to keep doing this? Are we going to do it in the same way? What's our mechanism for rolling the key?

So, there's a lot yet to be determined after lessons learned from this event, since it was only a week ago, or a week-and-a-half.

RUSS MUNDY: If I could, Wes, one more thing to add. The community definitely will have input on deciding when the next key roll is. This is something that the organization in ICANN – the organization CTO – is very keen in getting input from the community. So, if you have contributions, ideas, or thoughts, they definitely want to hear them. There are multiple mechanisms that will be put in place to seek those, but do think about it and contribute your thoughts.

WES HARDAKER: And, there will be, again, a presentation about the key roll coming on Wednesday at the DNSSEC Workshop.

JOHN LEVINE: Hi, Wes. I'm John Levine. I'm going to ask the same question I always ask, which is, I host DNS for about 300 zones for [inaudible]. For half of those zones, I'm the registrar reseller. So, I actually have direct access to the registrar. For all them, the DNSSEC signatures actually work.

For the other half, I'm a third-party DNS provider. With the best [inaudible] in the world, at this point, for all those other users, the only person who can make the DNSSEC work is the registrant, or the person with the registrant's credentials. So, either I have to walk my users though the process of installing a DS record, which is easy for you and

me but not necessarily for everybody else, or else I have to get the credentials for their registrar account, which I don't want.

I know that this is something that could either be solved at the registrar level by adding some way to be the DNS updater contact, or something we could fix in the IETF by having some sort of you – well, you know, some sort of automated thing.

Neither one seems to be making any progress. I realize that I'm not the most common case, but I think third-party DNS providers are not rare, and we need to fix it.

WES HARDAKER: Yeah. And, you're right that there is work to do be done there, especially for people that run a lot of zones. I did all of mine manually because I serve my own zones as well. And, there is effort. That's gotten better.

So, there's a few things I can tell you that have come about in the last couple of years. One, there's some providers that are creating a button so that, if you publish your DS record, you just go click on the key that says, "Is this the right one?" so you don't have to paste it in anymore. It'll actually go discover it for you.

Two, Warren Kumari – you ought to talk to him at some point, or [Oliver], who created the CDS record, which was a way to automatically update keys to communicate in the DNS between the child zone and the registrar.

But, I know –

JOHN LEVINE: Yeah, but it doesn't install it. Only updates it at this point. [inaudible]

WES HARDAKER: There's discussion about doing leap-of-first-faith and things like that, too. That actually has been discussed a lot in the last month: people are starting – why don't you come up, Warren? I should let Warren talk to that. But, people are actually now talking about pushing that out into production more. So, I'm finally beginning to see traffic of, let's implement this. Who has implemented it? So, actually, just very recently, that's come about.

You want to talk?

UNIDENTIFIED MALE: Yeah.

WES HARDAKER: Let Warren go first.

WARREN KUMARI: No, you can go first.

UNIDENTIFIED MALE: Yeah. I just wanted to comment that there are at least two or three countries these days where this is now live (pushing CDS records): the Czech Republic, Switzerland, and Lichtenstein. So, in those countries, if

you have a domain and you just display a CDS in your zone, then it's taken by the registry, and then the [whole] chain is set.

So, at least there are some attempts where this started and, of course, it's demonstrated that it's going to be a global initiative supported by dot-com. But, that's probably the beginning and something we can learn from. So, I think there will be a presentation about it in the DNSSEC Workshop this week. So, I invite you to join it. [inaudible].

WES HARDAKER: So, there is work to be done there, but the good new is that there's forward momentum, I think at this point. Yeah, and there's a panel on Wednesday, as he said.

RUDOLPH DANIEL: Hi. Good day. My name is Rudolph Daniel. I'm an ICANN Fellow.

WES HARDAKER: Welcome.

RUDOLPH DANIEL: I don't know whether it's appropriate, but KSK rollover was just mentioned. I just wanted to know – it's been successful so far, but have the old keys been erased, or they still in operation?

WES HARDAKER: Good question. Two things. It has been successful so far. The TTLs on the data (Time To Live) that are allowed to be in a cache was only two days. We are now ten days out? I've forgotten exactly. So – yeah. I can do math. Ten days. 21 minus 11. So, we are ten days out, which means that no DNS resolver should have kept it this long. They had to have updated their information, and they should have noticed a failure, unless they're running something very broken.

So, we are beyond the point where – anybody should have noticed a failure as now.

As far as removing the old key, there's two aspects of that. One, we still have the old key in the root zone, but it's not being used. It is sitting there still, but it is not being used.

On January 11th, it will be flagged. There will be a flag changed in it that will say, "It's now time to no longer trust it." It's called the revoke bit. In other words, you're revoking trust in it. That will be published for another three months, and then on April 11th, it will actually be removed from the root zone entirely.

So, there's more to the key rollover process. At this point, there is no expectation that we will switch back to using the old key, because it looks like it has been successful.

Okay. Good question. Thank you.

UNIDENTIFIED MALE: Hola. Is there any plan to integrate a notification to the user in the browser about the certificate, about if we are [inaudible] the site with the DNSSEC?

Another question. Okay, I will propose to my customer to use that .com and not a generic TLD because they are non-signatory in the generic top-level domains. So, it is more of a marketing promotion for other domains, and they need to go faster to implement DNSSEC.

My question is, do we have any plans for the user to show that it is the right site?

WES HARDAKER: Sorry, what was the last question? Any plans to show what?

UNIDENTIFIED MALE: In the browser. [inaudible] Okay, we can see the [inaudible] signature of any … I saw in the slide a notification in the browser, and I saw an image of the browser. It's integrated maybe in a browser extension or something like that?

WES HARDAKER: Do you want to speak to that? No? Okay. So, there actually is a browser extension you can install called DNSSEC … I'm blanking on the name. If you go search the browser extension for both Chrome and Firefox, there is one that you can install that will install in ICANN in your browser.

The browser vendors are not very interested in doing DNSSEC directly in the browsers themselves. There's a long history with that. You had to contact them and complain.

So, DNSSEC Validator – thank you, Tim. DNSSEC Validator is the name of it. If you go get it, it'll actually install it. Then, Russ and I, many years ago, worked on a browser that actually did DNSSEC down in the basic library of it. The browser is called Bloodhound. It hasn't been updated, so it actually doesn't have the new key, by the way. There was one usage of that on [Twitter.]

RUSS MUNDY: Yeah. We need to look at doing that.

WES HARDAKER: Yeah. So, that was actually modifying Firefox in particular to actually go and do DNSSEC straight into the bottom of the browser. So, there's more usage happening in e-mail and other stuff. Again, I'll talk about that on Wednesday, and you can see pretty graphs of the trend of e-mail actually getting more and more use with [inaudible].

BARRY LEIBA: On telling users about it, there's a lot of evidence over time that users don't understand what we're already telling them, and trying to tell users that have no idea what DNSSEC is something they don't understand on top of that is just not going to happen. There's no point in it.

WES HARDAKER: Yeah. In other words, the way that people are talking about doing it now is you just give it an error message and you don't ever show them a green or a red. You just don't let them go there. There' s a lot of evidence saying that, generally, end users don't have the knowledge to make a good security decision, so you don't let them make a security decision. You just deny them.

Behind you, Andrew.

UNIDENTIFIED MALE: Yeah, but just to comment on that, it doesn't mean that we don't need to engage and tell the end user and explain why that message is appearing. It doesn't mean it should be a banner or blocking the content. It would be more a little bit of an explanatory message, because, if it is the first time, okay, it's novelty. You don't know what it is. But, if the message is self-explanatory, in a sense, it could be a good way so that, the next time they appear, it would actually help, or …

WES HARDAKER: Yeah. That's a long, generic security debate about how much to let the users do. I, personally, being a technical geek, hope that they always give me an option to see the most detail possible. But, I do work with my relatives that don't understand stuff, and they call me. So, it's the full range.

Anybody else have a question?

UNIDENTIFIED MALE: I have one. We have many issues. We can store in the local host, and the host [inaudible]. If we can see a notification in the browser … I don't know if [inaudible] proxy, and we can get any other issue to have advance to have a benefit of the signature. So, it's very important to identify to the user and to have many issues [inaudible] of the Internet.

WES HARDAKER: Yeah. So, today, we're talking really just about names. If the proxy is also using the ISP's resolver and it's validating, then it'll be protected as well. But, you're right that there's multiple layers and multiple caches in HTTP-type mechanisms or in mail and all sorts of other stuff. All of those DNS lookups need to be validated to get a complete chain of security. You're right about that.

One behind you, Andrew. It's always behind you. Have you noticed that?

JOSE ALBERTO: Hello. My name is Jose Alberto. I'm a member of the ICANN Fellowship Program.

My question is related to Tor. [When we use Tor for some thing, does the DNS security apply to Tor?] It's applied to, in this case … I don't know. Is this possible? Or, do they use another kind of security in DNS?

WES HARDAKER: That's a good question, and that actually reminds me that I skipped the blockchain question earlier as well. There are alternate naming systems in use within the Internet. They're less popular one. Tor is one. There's actually a Namecoin, which is blockchain-based naming system. They're not compatible with the DNS. So, DNSSEC is really protecting just the DNS – the common naming resolution. That doesn't prevent these other mechanisms from being used. It doesn't prevent them from having their own security system.

But, no, I don't think the DNSSEC applies to Tor. I suppose that, in theory, it could, because it acts kind of like DNS. I'm not an expert on Tor, unfortunately. Anybody else?

Not enough to answer. Okay.

So, my guess that no is probably the answer, that it probably can't help there. But, you'd have to work with somebody that knows Tor better. Unfortunately, we don't have any Tor experts in the room.

It won't work. Okay. We found the definitive answer. No, it will not work to help.

FAN-CHIEH LIN: Hello. This is Fan-Chieh Lin from Chunghwa Telecom. I'm Jason, by the way. It's me, again. In my understanding, DNSSEC is designed to protect DNS transaction. I read an article arguing that DNS cookies are introduced by [RFCs, MTAs, MT3]. Is there to be a more lightweight method to do so? Could you share some comments on that argument?

Thanks. And, I hope I'm not deviating from this topic too much.

WES HARDAKER: No, that's just fine. So, DNS cookies solve an entirely different problem. So, I'm going to correct the very first sentence you said. It had a slight error. It's not bad, but you said it protected the transactions. DNSSEC doesn't protect the transactions. It protects the data.

So, let me give you an example. If I was going to Russ some DNS information and say, "Outside of DNS, I'm going to tell him my host name is at 1.1.1.1, and here's my signature," he can give it to the entire room, and it can verify all the way back. It's actually locking the data itself.

I don't care how it's transported. It could be over DNS. It could be over carrier pigeon. It doesn't matter. It's not the connection that's being protected by DNSSEC. It's the data itself.

That's important because of caching because, in DNS, it can take multiple hops sometimes. If you're using Google's 8.8.8.8 resolver, for example, they have many hosts that make up that system with a shared cache.

And, it doesn't matter who asks it and who answers it. As long as you can verify the data at the end, it doesn't matter how you got it.

The cookies mechanism was designed to protect a single transaction against, often, things that are too big, allowing you to get bigger answers from the server, in order to prevent denial-of-service type stuff.

The server may request that you come back with TCP or something that's not spoofable.

So, it's a different problem that cookies are handling. It does not protect the data inside the DNS transaction.

UNIDENTIFIED MALE: Hello. In my understanding, to secure the DNS data we have two kinds of mechanisms. The first one is DNSSEC, and the second one is DNS over TLS [inaudible].

So, usually I see that DNSSEC is very popular, more popular than DNS over TLS. So, can you compare between the two kinds of mechanisms? What is the most advantageous characteristic of DNSSEC compared to another? Thank you.

WES HARDAKER: They also have two different goals. DNSSEC, again, is to protect the data. So, it doesn't matter whether it went over TLS or not. It doesn't matter how you got it.

DNS over TLS is designed to protect the transaction between one person asking a question and where they are getting the answer from.

If your web browser asks your ISP, it can do that over a TLS connection to make sure that nobody in the coffee shop is looking at your request. But, you don't know if that the resolver at the ISP is able to ask the root and com and bigbank.com over TLS.

So, DNSSEC protects the integrity, whether you got the right answer. DNS over TLS is designed to provide privacy, to make sure that nobody else sees what you're asking and the answers that you get back.

It may be that, someday, TLS will be used everywhere, but you still don't know, because of multiple hops, if everything you got from Point A to B to C to D was actually secured. DNSSEC provides the security that, no matter how many hops it took, you know it's correct.

KEN HERMAN: Hi. Good afternoon. Ken Herman, independent consultant. Okay, I'm convinced of the value of DNSSEC.

WES HARDAKER: Yay!

KEN HERMAN: Can you say something about the level of penetration? How many people are using it? How can organizations, or even individuals, even know that the DNSSEC has established this chain of trust along the way?

Thirdly, can you say something about the cost to organizations for implementation? It might not be much for small businesses, which are dependent upon their ISPs, but perhaps larger organizations might have a higher cost. Thanks.

WES HARDAKER: So, come to the DNSSEC Workshop. The very first presentation at every DNSSEC Workshop, which will be on Wednesday this time – it's actually later in the week than it normally is – is a bunch of maps and a bunch of data for where the penetration is, what parts of the world are using it the most, and things like that, including: I'll give a presentation on the number of domain [inaudible] and how well DANE is getting deployed. So, there's a lot of technical information. I can't repeat it all there. [inaudible] [battery.] So, come to that.

In terms of … what was the second half of the question?

[KEN HERMAN]: The cost.

WES HARDAKER: Ah, the cost. It used to be much harder to do. If you ask your ISP to please turn on DNSSEC validation, they probably won't until people ask. Then, if they get enough people to ask, they will. These days, it's little configuration for them to turn it on, so they don't have a whole lot of excuse, other than they need to know how to debug it when it breaks. They need to know what happens when a key roll happens.

Most of that should all be automated these days. In the early days, it wasn't. You needed to know a lot more. It's very obvious now.

In the original days of being able to sign your zone, if you were going to do it yourself – you were going to host your own DNS, you were going to sign it yourself – that was a series of, like, 12 commands. We actually

documented it in an early deployment guide. It was very long and tedious. Russ and I worked together. Our colleague with ours came up with one tool. You run [zone-signer-space-file name] and it's done, and then you publish the results.

So, the tools have gotten a lot easier, so the cost is lower. But, there is no such thing as zero-cost security. There's always some security.

The other thing to know about DNSSEC is, if you're going to do it yourself, the DNS before used to be a publish-and-forget. You never had to modify your zone. You just go off and you publish it once. You could leave it for three years, and the data is still fine.

With DNSSEC, because it's time-constrained, you have to resign once a month, or depending on the parameters your pick. A month is common. I resign every two weeks for my zone, even though the security links are good for a month.

UNIDENTIFIED MALE: I have one remark to what you just said, because, with, for example, the Knot Resolver, which is doing the ultimate signing for your zone … if you have the system that publishes the CDS record, it's exactly how it used to be. Just publish it once, and all the automated mechanisms take care of the security. So, the cost is really minimal.

WES HARDAKER: Yeah. He brings up a great point. If you just tell your resolver – I think most of the authoritative resolvers actually have automatic signing

now – they'll just keep signing it for you. There's a bunch of stuff that makes it all automated.

I'm paranoid. I do it on my own laptop, and that's basically because I started in the beginning, not because I still should.

Any other questions? These have all been fantastic questions. Thank you very much.

THANH NGUYEN: So, I have another question. It's not a technical question. For the specific country of Vietnam – because I come from Vietnam –for example, in my government's own DNS system, it's [inaudible]. We want to change all the DNS system to DNSSEC. So, how much is the cost to transfer from DNS to DNSSEC, and how long? Can you estimate the time and the cost?

WES HARDAKER: That's a hard thing to estimate, so I'm not sure …

RUSS MUNDY: Let me make a few comments about that. I won't give any exact, specific answers but I can give you an idea.

So, one of the things that has been identified by people when they started to do DNSSEC and get things set up and running is that they often found that their DNS software has not been properly updated.

So, they may in fact be using, on their authoritative name servers, or, if they are also operating the ISPs and doing recursive name servers – that they may be three, five, or even ten years out of date. So, the very first step in most cases like that is to examine the state of your DNS infrastructure.

If it's good and up to date and the software is current, as we talked about earlier, the challenges of doing DNSSEC can sometimes be as simple as changing one or two option settings in the configuration file of the hidden master that does the actual distribution out to the authoritative servers.

It can be much more complicated if you want to set up a separate off-line cryptographic mechanism.

So, it varies, but the first and most important thing is to look at your infrastructure for DNS itself. Start from there. There's a lot information available online. The DNSSEC community has been incredibly helpful in sharing over time, and there's a ton of information available online for what else, on a country-wide basis, has been done.

WES HARDAKER: So, ISOC has taken over the website that you saw earlier, which was DNSSEC-Deployment, the one that had a picture of Steve Crocker on it. That's a great place to start for DNSSEC resources that can help lead you down the path of where to go. So, that's a great website that has a lot of information.

I have one called DNSSEC-Tools that is a little more geeky. But, you can go talk to your registrar, too. A lot of them have tools that, if you let them host your data and they run the DNS servers, too, they make it very easy to click a checkbox.

So, how much it costs depends on what you want to do. You have to go do that evaluation that Russ was saying. He said to start with what you're doing now and then figure out what that will cost from there.

RUDOLPH DANIEL: Hi. Rudy, again, ICANN Fellow. I'm just recalling now that I saw something called EDNSSEC. Is there such a thing?

WES HARDAKER: So, I think you're confusing two things. EDNS, without the SEC, is an extension in the DNS mechanism to add additional information when you're requesting information.

Actually, that's where you say, "I want to do DNSSEC. Please give me all the signatures and various stuff." So, that's an extension mechanism within DNS that DNSSEC needs.

RUDOLPH DANIEL: Ah, okay. Thank you.

WES HARDAKER: Any last questions? Barry?

BARRY LEIBA: So, if I have a minute, I'd like to riff on the early question about DNSSEC and HTTPS.

Initially, you didn't usually go to a website with HTTPS. You went with HTTP, and you got a redirect. An attacker could avoid you sending the redirect and fool you into not using HTTPS. So, you still needed verification there.

Then, we came out with something called Strict Transport Security, where the website, when you go to it, says, "Use HTTPs, and, by the way, always use HTTPS and never accept a connection to me without it." That uses a technique we call trust-on-first-use, assuming the attacker isn't going to intercept the first request.

To get around that, browsers are now putting a strict transport security list in the browser, so you don't even need to trust on first use. You never try to get to that website.

But, then you still get the rogue CA's that Russ mentioned, where, if you look in your browser and see how many root certificate authorities your browser trusts, it's hundreds of them, ranging from Verisign to some company you've never heard from in Taiwan.

If any of those gets compromised and is convinced to issue a certificate for bigbank.com, then you're compromised.

So, to get around that, we now have a thing called DANE, where the bank publishes its own records in its DNS to say, "These are the

certificates I want you to use." That gets around that problem, but guess what DANE requires? DNSSEC, because you're getting the certificates out of DNS.

So, it's all cyclic and it goes with what Wes said about everything being layered. You need to protect every piece along the way.

WES HARDAKER: The reality is that, if the user's first thing they put into a field in any application, web or otherwise, is a name, that name gets looked up. So, that's the first vulnerability point. So, there are other ways to get around all those types of problems, and they all have issues. But, if you secure DNS, then, actually, a whole lot of problems start going away.

That doesn't mean you shouldn't use HTTPS after that. That solves a different problem.

Any other last questions?

AHMAD ALSADEH: Again, Ahmad Alsadeh, ICANN Fellow. So, when I saw the sketch there, everyone in the chain has to have a certificate or verify the signature. If one is not deployed, then the DNSSEC will fail. This is right? Do I get it?

WES HARDAKER: Close. It's not that it fails, but you know positively that you are now going to a place that's not DNSSEC-signed, so you have to keep going and fall out of trust.

So, there's portions of the DNS tree that aren't signed, and your web browser will still work with them just fine, because DNSSEC actually hands you an answer. It'll say, "Well, here. You need to go to bigbank.com. By the way, they haven't deployed DNSSEC. You have no choice. You have to keep going with regular DNS."

So, there's this fallout mechanism, where, as the resolver is going down the chain, it eventually gets to a point, saying, "I can't do security beyond this, but you probably still want an answer anyway."

So, that's where we are today, where there's a lot of stuff that is signed and a lot of stuff that isn't. We don't give a lot of user visibility into that, aside from our messages.

AHMAD ALSADEH: Okay. Thank you.

WES HARDAKER: All right. We probably have time for one more question, if anyone wants to take one last stab at a question.

One over here.

UNIDENTIFIED MALE: [inaudible]. Do you get an error message in that scenario?

WES HARDAKER: Good question. So, the question is, do you get an error message in that scenario? Yeah, because what happens – well, you don't get an error message when you fall out into the insecure portion because that's just your resolver continuing to give you an answer. Your resolver will tell your application whether it got a secure answer or an insecure answer.

Right now, browsers or other applications don't really look at that bit of information because exactly how to present that to the user and whether that's a wise choice to present to the user is sort of up in the air in terms of whether that's helpful to end users or not.

What you will get is, if DNSSEC fails at some point, if Dr. Evil actually jumps in the way, a broken webpage message, saying, "That name could not be found," because the resolver tried. It tried to go look up the name, and if it never got a good answer because it kept getting bad answers and it never received a good signed one, you'll get the webpage that you get if you go to any broken name. It'll say, "I cannot find that name."

You won't know that it was because of DNSSEC protecting you. You'll just know that it failed to lookup the name for you.

Okay. So, with that, I think we will wrap up. Did you have closing remarks that you want to add about Wednesday?

RUSS MUNDY: Well, we have had several promotional – unintentional, but it's good – advertisement for Wednesday. We will have a series of panels. One of them deals with DANE. We'll have our introductory part that talks about

the maps that are created that chose the amount and type of DNSSEC deployment in various parts of the world, geographically.

So, we invite everybody to come back Wednesday. It starts at 9:00 and goes until, I think, 2:30? Kathy, is that right?

KATHY SCHNITT: 3:00.

RUSS MUNDY: 3:00. Okay. So, please do think about this. Thanks, everybody, for coming. We sure hope you were able to get your questions answered. If you want to catch any one of us from up here, or any of the players that have been up today, with DNSSEC questions, we're all happy to give you more answers and have one-on-one chats in the hall.

Thanks, everybody.

WES HARDAKER: Yeah. Thank you. So, one final point. On Tech Day, which is tomorrow, there's often a whole lot of DNSSEC-related concepts and discussion that happen there, too. There almost always is. I don't remember what the agenda is this time. But, usually a lot of the presentations are about TLDs deploying DNSSEC and things like that.

RUSS MUNDY: Right. And, that starts right after the opening ceremony at 10:30.

WES HARDAKER: Right after the opening ceremony. All right. Thanks, everybody, for coming. I hope you got something enjoyable out of it.

[END OF TRANSCRIPTION]