HYDERABAD – DNSSEC for Everybody: A Beginner's Guide
Friday, November 04, 2016 – 17:00 to 18:30 IST
ICANN57 | Hyderabad, India

WARREN KUMARI: So, we're going to give people another minute or two to walk in and sit down before we start.

Anyway, I guess we'll get started. Hello everyone. This is DNSSEC for Everybody. Usually, this is presented by Dan York and a few other people. Unfortunately, they weren't able to make it here this time. I apparently don't have power for the laptop, so hopefully it stays running the whole time. Oh, well.

Just out of interest, who here has been to a DNSSEC for Everybody session before? It's largely going to be the same. You can once again laugh at us when we do the skit. More people have because I had a lot more laughing then.

So people who know anything about DNSSEC probably think that it was invented in the last sort of 10, 15 years, something like that. It turns out that actually much order. It was invented around 7,000 years ago, back in the days of the cavemen.

So this is Ugwina. She lives in a cave on the edge of the Grand Canyon, and this is her boyfriend Ug. He also lives in the cave on

Note: The following is the output resulting from transcribing an audio file into a word/text document. Although the transcription is largely accurate, in some cases may be incomplete or inaccurate due to inaudible passages and grammatical corrections. It is posted as an aid to the original audio file, but should not be treated as an authoritative record.

the other side of the Grand Canyon. Unfortunately, the Grand Canyon is really big and it takes a long time to go all the way down, a long time to go all the way across, and a long time to go all the way back up. So Ug and Ugwina don't get to meet very often. This makes them sad.

So on one of the few times when they do actually get to meet and have a chat, they sit there and they notice that there is smoke coming from Ug's fire and they come up with a really clever idea. If they use smoke signals, they can chat to each other across the Grand Canyon and they wouldn't need to waste all this time going up and down, and back and forth.

And then, one day, another caveman called Kaminsky moves in. And for some reason, he likes to cause trouble. And what he does is he also starts sending smoke signals. Because Ugwina is far away, she can't tell who's actually sending the smoke signals, so she doesn't know which set of messages to believe.

So she gets annoyed by some of the things that she thinks Ug has been saying. She goes down the Grand Canyon, she walks across, she climbs up the other side, she starts shouting at him. And Ug says, "I didn't say those things." And eventually they figured out that Kaminsky has been lying, putting extra signals in.

They don't know what to do about this, so they go along to some of the village elders and asked them what they can do to help. One of the elders, Caveman Diffie, thinks that he might have a really good idea.

He runs along into Ug's cave and he gets a handful of this very special sand. And the thing that makes this sand special is it only exists in Ug's cave. And he takes a handful of it and he throws it into the fire and the smoke turns blue.

So now Ugwina can go back home and she and Ug can chat because she knows that all she needs to do is watch for the blue smoke. She can ignore any smoke which isn't blue and that way she knows it's actually coming from Ug.

And that's kind of what DNSSEC does. It adds a special blue smoke to the DNS packets, a way for you to tell which ones are the real ones and which ones aren't.

And now I'll introduce Wes. We made the introduction a little bit slower because Wes had to be somewhere else, but he's shown up.

WES HARDAKER: That is turned on?

WARREN KUMARI: Here, if you just sit down and use that [inaudible] unless you want to talk [inaudible] with mic.

WES HARDAKER: Good evening, everyone. I'll be emceeing through a lot of the rest of what's going to go on. So we are going to go on to sort of an introduction of the DNS and DNSSEC next, and give you an overview and then we're going to give you a little live skit to make it entertaining and fun.

So next slide, please.

So first off, the high level concept of how the DNS works is that there is sort of a tree structure and actually, we keep calling it a tree but to me, it's sort of upside down because it grows downward like the roots of the tree almost.

And we start at the top, which is the root and that's sort of the generic place where everybody goes where if you have no idea who else to ask, you ask at the root. So, for example, if you are trying to look up the name bigbank.com and you don't even know where com is, you have no idea where to start, you ask the root.

And then the root says, "Well, I don't know where bigbank.com is, but I do know where com is. It's down here." So you follow the tree down and we'll walk through that a little bit later in a skit that will make it even more clear.

Next slide, please.

So a resolver, which is usually running at your local ISP and is responsible for handling all that discussion for you. Your computer, your phone, or your tablet does not actually typically do all of that chain. It's just going to ask the question to your local ISP's resolver and say, "Where is bigbank.example.com?" And the resolver knows how to traverse the DNS hierarchy from the top all the way down to get you the answer and then it sends it back to you. Each level refers the resolver to the next level until finally the answer has gotten and it returns it all to you.

The other important thing to note is that the resolver caches that information for a while. So if you ask it again, it's going to answer much faster the second time because it doesn't have to go find out where everything is including once it knows where com is, it doesn't need to go ask the root again for that matter. It caches a lot of that information for a while.

Next slide.

So the one problem about the DNS is that it was invented at the time where there was not a whole lot of security put into the foundations of the Internet because there weren't too many bad actors back then. They were all connected by organizations that knew each other, so there was no security.

But now, names are very easily spoofed, meaning anybody can really return an answer for a name even if you're not the proper person that knows the information. And caches are easily poisoned. It's a little bit harder more lately due to some other security modifications other than just DNSSEC. But once you have a bad answer in your cache, you're going to continue answering it for a while. So once the resolver has a bad answer, it's going to keep feeding that bad answer until the cache runs out.

Next please.

So now, we're going to dive into a little skit/play. And I'd like my actors to please stand up. You can see that they're all wearing wonderfully cute T-shirts with different roles. You should put your wonderfully cute T-shirt on, Warren. That would be good.

And we do need the roving mics. These are actually different than the ones I was using all week.

So which laptop was controlling the slides because I still need to go forward? Thank you.

UNIDENTIFIED MALE: There's a clicky thing next to it.

WES HARDAKER: There's clicky thing. Excellent. We like clicky things.

So in the story that Warren just described to you there was Ug and Ugwina. Well, Ugwina is really the resolver. Ugwina is the one that's needs to ask the question, "Where is Ug?" Or where is bigbank.com? And Ugwina is confused. As you saw, there may be two different versions of an answer that she's receiving back and we'll get through that in a minute.

So let's start with the simple case. Sorry, the slides changed a little bit since the last time I saw them. And so we're going to start with a simple case of we're going to have Joe User. Raise your hand, Joe. Joe needs to do some banking. Joe needs to go to his bank [inaudible] he goes to the right place. But Joe is going to go to his bank and say, "I need to withdraw some money. I need to do some transactions. I need to pay some bills." And he's going to have to talk to his ISP in order to get answers. There is the ISP.

And then, to the far – so Cathy will be the root server. She is where this is all going to start, and then we also have com and we have bigbank.com to show an answer. And remember that in the end, computers only speak with numbers. They only speak with an IP address or an IPv4 or IPv6 address and it's the name's job to translate all those into numbers that everybody can speak.

So with that, I will turn it over to the actors.

UNIDENTIFIED MALE: Hello everybody. And today, I want to go do a financial transaction at the bank and I'm going to type in my browser, "www.bigbank.com," and let's see what happens.

I want to go to bigbank.com

UNIDENTIFIED MALE: I don't know where bigbank.com is. Let me go see if I can figure that out. Hey Root, one of my users wants to go to www.bigbank.com. Can you tell me what that is?

UNIDENTIFIED FEMALE: I do not know where it is, but I do know where .com is. He's at 1.1.1.1.

UNIDENTIFIED MALE: Hey .com, one of my users wants to go to www.bigbank.com. Where is that, please?

UNIDENTIFIED MALE: So I know where bigbank.com is and that’s at 2.2.2.2.

UNIDENTIFIED MALE: Big Bank, I would like to know where www.bigbank.com is. One of my users wants to go there.

UNIDENTIFIED MALE: Great. I love my users. Please tell him to go to 2.2.2.3.

UNIDENTIFIED MALE: Hey, User, I figured that out for you, 2.2.2.3 is where you want to go.

UNIDENTIFIED MALE: Perfect. And then, I go on my browser and I do my financial transaction, and everything is good.

WES HARDAKER: Excellent. Thank you all. We’ll get back to you in a minute for an additional round.

All right, so that was easy, right? That’s how the Internet should work. Everybody answers truthfully. There’s no malicious people. There’s no bad actors. And all works well and Joe goes off and does his banking, as appropriate.

Unfortunately, as we mentioned earlier, anybody can sort of spoof answers in the DNS. Wrong direction. So one of the things that can happen, as you can see in the right-hand box, the right-hand bottom box, the red box is that somebody else could answer who bigbank.com is. Somebody else could answer that just as easily, so we have to be very careful.

So now we're going to see a second version of the skit where there may be a bad person involved. And I will let the actors get back up and do that.

UNIDENTIFIED MALE: All right, so the same thing as before, I want to go to the my bigbank.com. I want to do a financial transaction to transfer 50 rupee from one account to somebody. I want to go to this address.

UNIDENTIFIED MALE: I don't know where that is. Let me go figure it out. Hey Root, one of my users wants to go to www.bigbank.com. Can you tell me what that is?

UNIDENTIFED FEMALE: I'm afraid I can't. But I can tell you where .com is. He's at 1.1.1.1.

UNIDENTIFIED MALE: Hey .com, one of my users wants to go to www.bigbank.com. Where is that?

UNIDENTIFIED MALE: I don't know where www.bigbank.com is but I do know where bigbank.com is. It's at 2.2.2.2.

UNIDENTIFIED MALE: I'll go try him. Hey Big Bank, one of my users wants to reach you. He wants to go to www.bigbank.com. Where is that, please?

UNIDENTIFIED MALE: Oh, great, I can help you with that. All you need to do is go to 6.6.6.6.

UNIDENTIFIED MALE: Awesome. Thanks. That was really helpful. 6.6.6.6 is where you want to go.

UNIDENTIFIED MALE: Oh, perfect. So I put that in my browser and then I connect to a bank. It doesn't look necessarily the same as usual. For some reason, my bank account is lower than usual. I think I got compromised.

UNIDENTIFIED MALE: Not my problem.

WES HARDAKER: All right, wonderful job. So, you can see – go ahead and give them a round of applause. That was fantastic.

UNIDENTIFIED MALE: We’re not going to sit down again.

WES HARDAKER: Yeah, no, don’t sit down. Stay standing because we’re going to see how DNSSEC fixes that, which is why you’re here. So the DNSSEC was invented much more recently than the rest of the DNS. It’s sort of some of security extensions to deal with this problem. And so what we’re going to show next is sort of what happens with DNSSEC. And the important thing to realize is that in DNSSEC, every record that is handed off is signed in such a way that you can cryptographically verify that nobody has modified it, and it’s signed all the way from the root on down. And the actors will go through and show that next.

UNIDENTIFIED MALE: Now, I know that DNSSEC exists. As a user, I turn DNSSEC validation on and make sure I only accept authenticated queries So, again, I’ll type in www –

UNIDENTIFIED MALE: You sign first.

UNIDENTIFIED MALE: Oh, sorry. So now, we have the root, .com, big bank TLD, they are all signed with DNSSEC, that's the color they have, orange and yellow. And this is a procedure they do each TLD, each ccTLD, to create digital signature for all the answers that they provide.

So now, I'm more confident in doing my DNS, my big bank, my financial transaction. So I want to go to www.bigbank.com. Please get me there safely.

UNIDENTIFIED MALE: You really like going there, don't you? Hello Root, one of my users wants to go to www.bigbank.com. Can you please tell me what that is?

UNIDENTIFIED FEMALE: I'm afraid I'm not sure but I do know where .com is. He's at 1.1.1.1 but you know what, I really need to sign this first.

UNIDENTIFIED MALE: Hm, let me check. Yeah, that will check the same color. Okay, I believe that. Hey .com, one of my users wants to go to www.bigbank.com. Can you tell what that is?

UNIDENTIFIED MALE: I don't know where www.bigbank.com is, but I do know where bigbank.com is. And while I tell you, it's 2.2.2.2, have a signature too so you know it's for real.

UNIDENTIFIED MALE: Let me just check that. Yup, that's good. Cool. Let me try bigbank.com. Hey bigbank.com, one of my users wants to www.bigbank.com. Where is that, please?

UNIDENTIFIED MALE: I can help you with that. It's at 6.6.6.6.

UNIDENTIFIED MALE: There's no signature in that. Who the hell are you, dude? Seriously.

UNIDENTIFIED MALE: Let me help you with that. It's at 2.2.2.3 and here is my signature.

UNIDENTIFIED MALE: Woo-hoo, excellent. Hey user, 2.2.2.3 and I checked all that, I've got signatures, it's all good.

UNIDENTIFIED MALE: Oh, thank you. Now I feel way more confident in going to the bank and doing my transaction.

UNIDENTIFIED MALE: So you should.

UNIDENTIFIED MALE: Thank you.

WES HARDAKER: Absolutely. Thank you very much, wonderful guilded actors of ICANN ,I guess.

So from there, we will go on and we'll talk more in detail over the next remaining 40 minutes or an hour about exactly how all that works and then go into greater detail about some more of how this actual system works.

But one important thing to realize is that DNSSEC protects DNS. When you go ask a question to the DNS system, it's going to give you an answer and you can verify that that answer is true. That doesn't mean that's going to protect all of the Internet. We're only protecting the DNS system.

There are other technologies for protecting other elements, like the routing system has its own routing security technology that's being developed. HTTPS is the TLS version of HTTP to protect that. So we're just trying to protect with DNSSEC how you get the correct answer. Not necessarily the rest of the infrastructure that goes along with your browsing experience, for example.

So DNSSEC solution is the solution to providing these answers. It's been in deployment for the last decade-ish and it's getting increasingly deployed. The root has been signed, I don't remember the exact year, but it has on the order of five to ten years, 2010? Okay, roughly 2010.

So, in all of the – many of the TLDs below that are signed and a good percentage of even .com is signed and some zones are even more fully signed. So the level of deployment is going up over time.

The keys and the signatures are there to verify the information, and you can get the keys for everything below the root just by asking through the DNS, so it's actually all those keys for how to do the verifications of all the steps along the way are also in the DNS.

The DNS system is a look-up system and then you can simply ask the DNS for the key that you need. The one key you do need at the top is the root key. Once you have that, everything else you can find securely.

A resolver knows what that root key is. A resolver has to know ahead of time when it starts up, it's preconfigured. Here is the root key, as long as Cathy the Root matches the answer everything below that, you can figure out, including the keys for bigbank.com and .com. There's a chain of trust is what we tend to call it from the top down.

So this is sort of the same things what just happened that you saw visually with the actors, right? The checkboxes in this diagram show the legitimate things that you can verify and you can go through and say, "Yup, that one was true. Yup, I got the answer from .com, it was signed properly." And if somebody tries to return an invalid answer for Big Bank, you can very quickly tell it was not true. You can check that cryptographically.

So we did that skit already, so we're going to go on to an example of why you need DNSSEC and a simple guideline to getting it deployed.

So why worry about the DNS? So one thing that you all know is that you would hate to memorize an address, an IP address for every bank that you wanted to visit, every website you wanted to visit. Telephone numbers are hard. How many telephone numbers of your friends do you actually keep in your head? You don't. You have a phone that has a contact database in it because you know their names. It's very hard to remember numbers. To be honest, I don't know the numbers of a good numbers of my friends because I always use my contact database. The DNS is the same way. The DNS helps us connect the way we think (names) to the way computers think (numbers), and the Internet addresses are the numbers.

The one thing that's important is that all applications sort of require the DNS to work for humans to interact with them. And if the DNS doesn't work right, then the applications don't go where you're intending them to go.

So the problem with the hijack threat, the skit that you just saw, is that people can sort of redirect you and gets you to go to the wrong place by giving you the wrong address. There's more than one way to get you to get redirected and the Internet Engineering Task Force is trying very hard to fix all of those different layers. This again is the DNS side. We want to get you to the right address to start off going to and that includes the ability to protect you from Dr. Evil stepping in and handing you the wrong address when you went to look something up.

Because what can happen if they gave you the wrong address? You could go to the wrong website, where you might type in your bank’s password. You might send your e-mail to the wrong place, a “man in the middle” attack, or you might go to the wrong webpage that has malicious activity and malware on it.

It’s actually quite easy to find DNS tools on the Internet that will let you do “man in the middle” attacks. So it’s very important that we really push forward within the Operational Committee and within the ICANN communities to really push this technology forward to make sure that we get rid over this problem entirely.

So how does DNSSEC help? Again, as we’ve explained, DNSSEC ensures that you’re getting to the right location. More importantly, you’re getting the right address to get to the right location. And the hijack will be prevented entirely by the signatures that were represented by the colored badges on our actors’ badges.

Excuse the technical malfunctioning going on in front of me. Power issues, I guess.

So let’s think about that same sort of scenario that went on from a graphical representation. We’re going to give this to you multiple times, because we want you to get the idea into your head of exactly what’s going on. It’s very, very hard. It’s sort of a complex sequence and we'll show you how complex it is again in a few slides.

But this is sort of the diagram for what just happened. When an initial request comes in, it changed through to a whole bunch of other servers. And you can see just by the minimal number of arrows that we have on this diagram that there's actually a lot of transactions going on. And most people don't realize when they type in www.bigbank.com that there is actually a whole lot of transactions. Everybody thinks that it's just one DNS request going out. But it's not. There's potentially lots of servers involved. And we'll see in a slide in a little bit that shows actually how many DNS servers are involved with, say, a simple webpage, which also has images and java script and other things associated with it from different places.

In the end, we only want the right answer to get to Joe User, which is in the bottom left corner. There we go. We actually have a webpage called dnssec-deployment.org. And on this webpage, it tells you all sorts of information about how to get started deploying DNSSEC both within your own infrastructure in order to how to validate queries that are going on in the world that have been signed, as well as how to get your zone signed itself.

If you run a zone or you own a zone, how you can go about signing it yourself? Some registrars will do it for you. There's tools available on this webpage if you want to do it yourself. And then there's tools for how to deploy a resolver that actually knows how to speak DNSSEC and do validation.

On this webpage, you'll note that there's a green checkmark in that upper left corner there. That's actually showing you if you go to that webpage and there's a green checkmark, you'll find that you are doing DNSSEC validation. If you go to that webpage and you see this yellow diamond, it means that your system and your ISP is not actually protecting you from DNSSEC.

Now, DNSSEC can actually be implemented at various places. It's most frequently done at your ISP's resolver and it will return either good answers to you or bad, so it will either protect you from Dr. Evil or not. You can also do it on your own laptop. That's a little bit harder. I do, but I'm a technical geek. Typically, we want to encourage the resolver owners, the ISPs, to actually deploy it in your local ISP because then it protects everybody within your ISP.

So this sort of the same diagram that we saw before and I'm not going to trace down all the arrows as they come very quickly, as you'll see. But in the end here, we will see that Dr. Evil down below actually returns an answer faster than anybody else. You know that Dr. Evil is very close to the source that ask the question. All he has to do is beat the rest of the infrastructure. If

Dr. Evil can return that answer faster, he wins. Because without DNSSEC, you believe the very first thing that you heard back regardless of whether it came from a good person or a bad server.

So the red lines [inaudible] came back in this diagram and came in slower than Dr. Evil's hacker, which is why the user believed the bad response.

So this version is sort of the same. But now we see that the bad answer from Dr. Evil has been blocked because even though the answer got there first, it was checked by DNSSEC, it was checked with cryptographic signatures that showed, "You know what, this one isn't right," and it just drops it on the floor and says, "I'm not going to take this one. I'm going to keep listening. Let's see what else comes along." And this is exactly what happened in the skit before, where the final answer actually does get to Joe User.

So that's actually what's happening with that green checkbox is that the first answer does come back but it gets there faster, but then if you have a web browser that actually knows how to do this or your local ISP is doing it for you, that bad answer is going to be ignored.

This is actually a spoof that we did in the last company I worked for where we actually inserted a fake article into the webpage using just DNS spoofing. So the first page with the green checkmark doesn’t have the picture of Steve Crocker in it because we spoofed his image because he thought it would be funny.

So remember before how I said your web browser produces many, many, many questions? This diagram, I actually created probably ten years ago now, to be honest. Each one of those lines is a DNS question or a DNS answer just from loading the CNN webpage. It’s that crazy and it’s gotten worse. I mean, I did this ten years ago and there’s a lot more lines now. And so you can imagine, you’re trying to make sure that every single one of those lines is being protected to make sure that your browser is actually going to the right place.

DNSSEC is a very complex system. There’s a lot of questions going on and most users are completely oblivious to how many DNS queries go out everyday just to do the simple things like send e-mail and look at webpages, and talk over messaging applications. There is a ton of DNS questions going on and this is another – the sort of version of the same diagram.

So some basic facts about DNS. It just provides a translation from names to network addresses. But as we learned earlier, actually, the DNS can also question other things. You can ask for keys. You can ask for the inverse names. You can say, “I have the number. I need to get the name." You can ask for, "Who's the mail server," which is independent of, "Who's your web server." There's a bunch of other things. It's just a key lookup kind of system.

And the important thing is it's the data that matters. When you ask the question, you want to make sure that you get the right answer for the data. You actually don't care how many people handled it before you. And that's one of the wonderful things about DNSSEC is that it protects the data.

So if I gave the answer to the first person over here and it went all the way back through the room all the way. And you played that secret listening game that you played as a kid where you whisper something in somebody's ear and it goes to the next person, and eventually it comes out the end and it's totally different. That can't happen with DNSSEC because no matter how many people touch it or how many people hold on to a copy of it, if it takes a year to get all the way through the room and back to me, I can still verify that it has not been modified since it was originally created.

This diagram actually I also created a long time ago. That just shows all of the different steps that happen when you are creating and publishing DNS data. The person on the left-hand side says, "I need to create a www record. How do I do that?"

Well, they add it to the zone data, they publish it to the authoritative server, the server that's responsible for returning the answers. The recursive server, the box, the lower left box sends the request to the authoritative servers to get the answer, and the client on the right sends its request to the recursive server to start the whole chain.

And they're numbered, so if you want to look at the slides later, you can find that the client sends the request. The request then goes up to the authoritative server, the answer goes back to that one and then over to this other box. And that's just one transaction compared to the thousands of lines you saw before. This is just one answer.

So DNSSEC works – I'm going to skip some of these slides because we're going to do some extra stuff at the end that we'd normally don't do as an added sort of bonus for you tonight. So I'm going to skip a few of these just to skip forward and hit the important facts for you.

One important concept is that you need to protect the zone data just as much as you protect DNSSEC. So if DNSSEC protects your data, well, what happens if you put bad data in? DNSSEC will happily tell you that the bad data is correct. The bad data has not been altered since you made the typo and put it in there in the first place.

So a lot of people concentrate on keeping the keys, their private keys for DNSSEC very safely kept but they fail to actually protect the people putting in the data and they leave the system of actually generating the data insecure. So you might have to remember that not only do you want to protect the security in properties associated with DNSSEC but you need to protect your zone data just as much.

So this is sort of the same diagram as you saw before but it only shows you the pieces that DNSSEC added. So the little lines, I'll point to the new pieces. Before, we added the record to the zone data and now we're going to sign that to get to the signed data, and the authoritative server is actually going to publish the signed data as opposed to the original zone data.

And there's also the facts that validating recursive resolver has to be able to get to the keys and it's by the validating recursive resolver knowing the keys of the root and being able to chain it all the way down that they can compare that against the signed data. So that they can see the little lock icon represents these signatures on the signed data that the validating recursive resolver can verify.

This is another one we're going to skip for today.

So, Julie, so I guess we have questions and then – we want to do questions before the follow on.

JULIE HEDLUND: Actually, Rick will be next.

WES HARDAKER: Rick will be next.

WARREN KUMARI: Rick? Uh-oh. Well, while we wait – well, there is Rick. Run Rick run. No, just calm down, it's okay. We have time.

RICHARD LAMB: [Inaudible] here?

WARREN KUMARI: No, anywhere you want.

RICHARD LAMB: Hello. Hi. I work at ICANN. I was one of the guys at the very beginning that was told – can you switch to my presentation? I was told, yeah, we're going to do this root signing thing and we've never done something like this before. And, gee, how do you make people trust you?

So sometimes that could be hard if you're ICANN. I'm not hearing any laughter, so, okay, that one didn't go very well. All right, anyway, so, as you heard from this lovely presentation these guys did and the skit they did, which I think is kind of funny, there's one key that's kind of important in this whole picture.

So how do you make someone trust that one key? That's a tall order. First of all, if it's us, it's like, "How can I trust this, really, really trust this?" So, we developed a system that had 21 people from around the world, 18 of which are not American, very important, that all hold physical keys, hold cards, hold various pieces of things.

And so they come together four times a year, something called the Key Ceremony. It sounds very strange and cryptic. But it really isn't. In fact, it is a turn of art and it's a common process used for certificate authorities. Anyone see the little locks on the side of their website when they go to an SSL page or secure page? So we borrowed from them.

We're not going to reinvent the wheel. We borrowed all these [inaudible] documents and things like that to develop from them, and that was back then in 2010. It was a community effort clearly and ICANN was involved, even Verisign was involved with it and we did it all at the behest of the community.

Where is the slide? The slide clicker? Well, thank you. Thanks. Does that work?

UNIDENTIFIED MALE: Mm-hm.

RICHARD LAMB: So what's this presentation about? Well, it's been six years. We're going to change this key. We're going to recreate another key. Very exciting for some of us. But the problem is if we change this key, as was explained earlier, that is embedded in and is part of resolvers and much of the network. And so if just change this without telling them, everything fails. Then we look really bad.

So part of the reason for me giving this presentation and part of the reason why they gave me a little timeslot here was to basically raise awareness to let people know that this is going to happen. Do class exercise here, whatever.

All right, so, just like I said earlier, there are maybe some strange terms like Key Ceremony and things involved here, but it's a very open process. We, ICANN spent many hundreds of thousands of dollars trying to learn some of the secrets of these processes that are used by certificate authorities and others. But since we're ICANN, it's all open, we tell you everything, we publish all the documents, you can copy this stuff, use it, do whatever. We'll even tell you where they are. These are – you can see, there's the laser.

All right, X marks the spot. This is right near the LAX in Los Angeles and there's a mark exactly where you'd want to drop the missile. And this is a place outside of 25 miles right outside of the nuclear blast zone out of Washington, D.C., not for any particular reason other than that's a really good data center. I wonder why. But it's a good data center. So we put our facilities there as well.

Next slide.

Just to show you, this was a very community-oriented thing. The name Kaminsky was chosen for a very particular reason. [Inaudible] there he is there and as an old saying, he's not an enemy shall we say. He's a [inaudible] critic. [Inaudible] your enemies.

UNIDENTIFIED MALE: [Inaudible].

RICHARD LAMB: Or I can just yell. Anyway, so, it was –

UNIDENTIFIED MALE: Hold it farther away.

RICHARD LAMB: Hold it farther away? Okay, fine. All right, so, you see here Vint Cerf, the real of father of the Internet, was involved in this as well. We have Anne-Marie from Sweden. We have people from all over the place.

Anyway, this is just to show you, yes, this is something that people get involved with. And in fact, I understand in the near term, maybe in the next few months, we're going to be looking for a few more people of that 21 people from around the world.

And as you can see, I've been told this a couple of times. As you can see here, it's very male heavy, so I get my hand slapped for that. I think it would be great to have some more women involved in this process. And the requirements are that you're technically active in the DNS community and that there's geographical diversity. So we're not looking for politicians. We're not looking for people that are trying to do something else.

All right, I'm sorry I'm wasting a lot of time there. This is a newer crowd so I thought I'd go into that a little bit.

So why are we changing the key? We did this in 2010 and it worked just fine. Well, part of it is good cryptographic hygiene. So we use something called RSA key. It's 2048 bits. It's probably good for another 30 years. When I said good, I mean, it would probably take another 30 years for somebody to break this key, to factor it or whatever.

But you never know. Algorithms change, discoveries are made on how to break keys. So it's a good cryptographic hygiene. But to me, the most important thing is the second one. If we don't ever do this, we won't know how to do it if we have to do it.

And the last point, we said we would do it. As you all know now that you've been at these ICANN meetings, ICANN has absolutely no statutory authority at all over anything as far as the root goes. It's a popularity contest. People trust us because we do what they say. We follow through. And so, we promise that we would change this key in five years, [inaudible] too much in that but over five years and so we're doing that.

All right, so, who's it going to hurt? Well, so far, DNSSEC, everyone here, we would love if DNSSEC was deployed on everything but it's not. But about 15% of the world's users sit behind a resolver that does DNSSEC validation that actually does look at DNSSEC, 8.8.8.

How many people here have seen 8.8.8.8, raise your hands. All right, good. All right, that's Google. Three guys in Manhattan. They're just sitting around and they decided to just throw DNSSEC into their resolver. But my God, I am so impressed. A very large percentage of the world uses that and I'm so grateful to Google for doing something like that.

Anyway, so it could affect that. If we screw up, it could affect that. Naturally, we're in contact with Warren and others at Google to make sure that they see the new key and it gets installed.

But if we misconfigure this, a lot of things are going to go bad. And as you saw from the skit, if the DNS does not respond with the right answer or with an answer, you get nothing and it looks like things are down and then they start calling people up. They don't call us up. They probably call up their ISP but they started calling a lot. As far as they're concerned, the network is down, the Internet is gone. So it's very important to do this carefully and slowly, and that's why we've taken a long time to do this.

Here are the documents – I'm almost done babbling here. Here are the documents if you guys want to look at it. Here's our plan. Anyone interested, please take a look at them. We have fallback plans of course in place should something go wrong, but I don't think anything is going to go wrong. I actually think this will go fine.

Oh, yeah, and just the other day, this is the 27th of October. We generated the new key. It's not in the Internet yet. Like I said, this is a slow process. We're going to take very small steps to get to this. But we generated the key.

And, yeah, this is the picture I was criticized at, very male heavy, so this is bad. Anyway, so what we do just like the first time is when the key is generated, we have a page that actually has something called a hash, a representation of the public part of this key and we all sign it.

And there'll be a second half to this in Los Angeles around the 2nd of February. So if you happen to be in Los Angeles, please ask and we can put you on this. This Key Ceremony is open to anyone. And please ask, you can sit there and watch the process and it will be boring but sometimes we have the good dinner afterwards. Like ICANN. ICANN throws a good party. Hopefully, we throw a good party.

So, all right, so these upcoming days, this is the stuff where you've got to start worrying. This is where it starts showing up on the Internet. It shows up in your DNS. September 19th, you'll see an increase in size. New keys will be introduced.

But October 11th is your key date. Yeah, I know, wait, laugh here. No, no one is laughing. All right, this is your main date here, October 11th, that's when the new key will actually flip in. All right, so that sounds like it's a long way off but we need to start pounding the drums so that everyone knows that this is going to happen.

I think the big guys will be okay, pretty much, but it's always going to be somebody on the fringe that's going to not update their key. There are automated processes for this. This is my favorite picture. You can pull this presentation down later but this gives you all the details of when we're going to do things. All right, very important but I just put it there just for reference.

So what do you guys got to do to do this? Well, I understand 8.8.8.8, yeah, Google, they just want – what's the new key? They'll configure it. They're smart guys. They know exactly what they're doing. And a lot of the very large ISPs may be following that pattern. I don't know.

But there's also an automated process in this [inaudible] at using numbers but the IETF comes up with various standards. There's a standard called RFC 5011. We're going to abide by that. So any system that's up following that standard will automatically see this key and automatically update it. There are other standards or an automated approaches we have in place that will also allow this to happen.

This is interesting. I mean, these slides are written by a gentleman, a colleague of mine named Matt Larson and Roy Arends at ICANN. So he put this thing in here. What if something screws up and doesn't work well? This is negative trust anger management, so trust anchor management – I probably said anger – trust anchor management.

So, anyway, so we're really looking for belt and suspenders here. I think that's pretty much it for me. Yeah, we have some test beds here. There's something written by our own Warren Kumari here, keyroll systems. I've written something as well, pretty pictures, things moving around. These are things for guys that are actually either resolver operators, hackers, I mean, people into this stuff that actually want to test their systems.

I don't see it here but ICANN is also going to have one that is running in real time. These are accelerated test beds. So there are things moving much quicker than in real time if you want to test stuff, a little more complicated really heavy geeky stuff. But hey, richard.lamb@icann.org, give me or send me an e-mail and I'll help you.

And we've talked to a lot of the vendors as well, with Microsoft for example, a lot of large vendors, so we're good here.

That's it. Yeah, there's this Twitter thing and yeah, I'm not very social. I don't know anything about this Facebook stuff or Twitter stuff. But there you go. That will eventually reach me. E-mail is really the best way to reach me and if I don't answer right away, just hit me harder and I'll come back.

Anyway, that's my requisite presentation with key rollover. And any questions? There's one on – any questions online.

JULIE HEDLUND: This question came in just as you started talking, Rick. So we thought we'd give it a priority. This question is from [Afifa Abbas] from Dhaka, Bangladesh. The question is, "Can you explain the use of zone signing key and Key Signing Key in DNSSEC?"

WES HARDAKER: So, let me thank Rick for his wonderful presentation and feel free to take a seat on our panel, Rick. But thank you very much. I appreciate it.

That's okay. That is an additional presentation that we don't normally give during this but the upcoming events are rather critical, so we decided to include that information.

That question is really one for the panel At-Large, so if anybody on the panel wants to answer it and again, we'll take questions from the floor. If you have questions about DNSSEC at all or if you have questions about any of the presentations from the [inaudible], now is the time to answer. Raise your hand and I will come with the microphone. In the meantime, do somebody want to answer that question? Warren?

WARREN KUMARI: I can take a stab and others can make it more coherent. So yeah, DNSSEC has both a Key Signing Key and a Zone Signing Key. The root key is a Key Signing Key. Basically, its sole purpose in life is to sign other keys. It signs the Zone Signing Key. And then the Zone Signing Key itself signs the information in the zone.

And the reason that there are two different keys is the Key Signing Key, you don't use very often. You only use it to sign the Zone Signing Key and that means you can keep it much more secure. You don't need to keep it around the neck. You can lock it up in a safe somewhere and that way you only need to have the Zone Signing Key available to actually sign the zone.

So you sort of now have two levels of security. You can keep the Key Signing Key locked away, the Zone Signing Key, you can actually use to sign the zone. And when you need a new key, you just use the Key Signing Key to sign the new Zone Signing Key.

I have no idea if that made any sense because I used the word key many, many, many times. I'll let someone else try it.

WES HARDAKER: Let me ask you a follow-up question Warren, which is that, can you name a zone that might need to frequently add new data so much the Zone Signing Key has to be constantly online and constantly around whereas the Key Signing Key can sort of be kept in a lot box in case they ever need to change the Zone Signing Key?

WARREN KUMARI: I can name many zones. One of them would be .com. There's new information added to .com every minute or so. And so, you need to constantly be signing .com zone with the Zone Signing Key.

There's also something to make it even more tricky. With certain cryptographic stuff, the more times you use a key, the easier it is for an attacker to try and factor that key. So that means that it's a good practice to roll the Zone Signing Key fairly often especially if you've got a large zone and you use it a lot.

That means that you should change your Zone Signing Key fairly often. It's kind of an annoying process. So if you manage just to have a Key Signing Key, which then signs the Zone Signing Key, my goodness me, I'm saying key a lot, you don't need to worry quite as much about dealing with the annoying part of changing keys.

WES HARDAKER: All right, did you want to add something before we could go on the next question?

UNIDENTIFIED MALE: I was just going to add on another operational reason for having the two keys. Because once you change the KSK, the Key Signing Key, you saw us a do the little handshake, you have to actually tell the level upstream. So for bigbank.com when it changes to KSK, I need to tell .com that there's a new key. So when it's more work to change the KSK than it is to change the [inaudible].

WES HARDAKER: Fantastically important answer. Thank you very much. Yes, that's absolutely true. So we're going on to the next question.

[AWAL]: NextGen, I am thinking that most of the work regarding the DNS deployment are actually technical and mostly involve the ISPs and also the DNS operators. So as a user, if I find that my request is not to the DNSSEC or not using the DNSSEC, as a user, what can I do then?

WES HARDAKER: Warren.

WARREN KUMARI: So what you can do, which is really tricky and annoying unless you're a crazy geek, is you could run a validating resolver on your own machine. That's a lot of work and that's kind of tricky.

Another option is you can change your DNS resolver to one that does do DNSSEC validation. So if your ISP does not currently do DNSSEC validation, you can ask them to please enable DNSSEC validation. If they don't, there are other resolvers like the Google public DNS one 8.8.8.8 that does validations so you could just change your resolver on your machine. You could reconfigure your machine to point that instead.

Verisign also runs a public DNS survey. I think it's 64.64.6, I can't actually remember the number. And then Open DNS, there are a number of other open resolvers that you can just use instead.

WES HARDAKER: And I just like to follow that up with, oh, look, we have a human that can't remember the sequence of numbers. That's why we have the DNS in the first place.

So, before you, there was somebody else that had a question back here, does somebody else that had their hand up before? You had your hand up before?

UNIDENTIFIED MALE: Yeah, I'm raising my hand.

WES HARDAKER: Okay.

UNIDENTIFIED MALE: From Egypt, second time fellow, the first comment I have a comment and a question. As the first comment, I use still the DNS, so my two keys, the ZSK and the KSK is secured. So I think that it's useless for me to do KSK rollover. This is the first comment.

Second thing, if whenever I can do a KSK rollover, do my [inaudible] show updates to trust the [inaudible] other side?

Third and the last here, third comment, DNSSEC doesn't secures our link between me and my [inaudible] and I don't trust to have any resolver locally at my machine. So is there any advice regarding that?

WES HARDAKER: Thank you for the question and that last part was something called – we called the last mile problem and that how you get a security answer all the way back to your laptop or your iPad or to your iPhone.

And, Warren has his hand gratefully up again.

WARREN KUMARI: So I can speak just to the last mile problem. For the last mile problem, if you're worried about making sure that nobody tampers with the response from your ISP to you, there are two options. One of them is to simply run a validating resolver on your laptop.

There is a tool, DNS Trigger I think is probably the easiest one, which is an application you can install and then it will make sure that it does the actual validation on your own computer, so you can then make sure that the data is DNSSEC secured.

At some point, there's also – the IETF is also working on another solution, which actually encrypts the information from your laptop to your ISP. It's a working group called Deprive and we're trying to deprive the attackers of the ability to make changes.

It's sort of main motivation was also to provide privacy to your queries. If there's an attacker watching your queries, they can figure out where you're going. This aims to encrypt the DNS so that attackers and [inaudible] as well can't see where you're going and so can't block you.

WES HARDAKER: A quick clarification, it aims to encrypt the question respond to only to the resolver or the people – the live data being sent, not the data being stored that's actually in the DNS I guess. Go ahead, [inaudible].

NAVEEN TANDOM: Hello, from India, ICANN fellow. A couple of questions, the reason is like why did you choose three months' time for the KSK [inaudible]? Any particular reason?

WES HARDAKER: Rick.

RICHARD LAMB: That's kind of a – I'm trying to remember why we did that. So, there are two reasons. The ZSK, originally, there are two keys. The KSK is 2048 but the ZSK is a little shorter. The shorter the key, more likely to compromise, so the more often, the ZSK needs to be changed.

So the ZSK now gets changed four times a year. Each one of those Key Ceremonies is a new ZSK. That's reason one. So we didn't want to – if someone asks you how good is a 1024-bit key, it depends on who you ask. If I ask with [Phil Diffie] and others of his colleagues, and I have. I get five different answers. Some of them say, "Yeah, you know I've heard," and they say, "maybe six months," for something like this.

That seems unlikely. But if six months is a lower anecdotal bound, you try you want to do something more often than that. That's reason one.

Reason two I a little bit more political. So this management of root key is done between ICANN and Verisign, two organizations. So the idea is that the KSK signs the ZSK. We don't want to sign a a large set of these for the next ten years at once because what if we don't get along?

So there's a little bit of a policy reason there, too. So there was a fair amount of thinking but that's probably more than you wanted to hear. But the main reason is cryptographic.

WARREN KUMARI: So one quick addendum from –

NAVEEN TANDOM: One more question, do you still trust the HSMs which does took a close? It still follows the 142 level 3 – do you still – I mean, at least on the idea of community.

RICHARD LAMB: I'll answer that real quick. So you said, do I still trust it? Well, what choice do we have? So there's [inaudible] level 4. Yeah, I'll leave it to Warren but there's basically a standard that we follow and because the idea is we're good bureaucrats and we want to follow a standard of security but that [inaudible] level 4 is about to. I think there's something called ISO, and the ISO Standard that soon will replace that and as soon as that happens and as soon as we can change, we'll change.

WES HARDAKER: Warren.

WARREN KUMARI: So yeah, I was mainly responding to Rick's what choice do we have. So at the moment, there are a fairly small number of people making HSMs but they are used to secure the most important cryptographic keys anywhere.

There are a large number of people who've been trying to attack HSMs and it turns out in general they're really good. However, many of the HSM vendors are based or have been based in the US or other countries where potentially people might not entirely trust that.

There is currently a project, which is called [Cryptic]. It's a global project that's trying to get people from as many countries as possible to contribute. And it is designing an open source, open design, fully open process HSM. So that way, anybody can validate the design and eventually anybody can build their own

HSM and know that the one that they've built hasn't been tampered with.

But it's a large project. It takes many years. HSMs are expensive [boxes] for a good reason. There's a lot of security around them but the HSMs at ICANN is using are, as far as anybody can tell, one of the best set of HSMs. They seem secure.

WES HARDAKER: All right, thank you very much. That was a good, great question. Can we go on to the next one, please?

UNIDENTIFIED MALE: Thank you, sir. From Afghanistan, are the resolvers are open to the entire world? This is my question.

WES HARDAKER: So are you asking are all the resolvers, are there some resolvers that are open to the entire world?

UNIDENTIFIED MALE: Yeah.

WES HARDAKER: Good question. Warren, you probably have more validity in that one than anybody else here.

WARREN KUMARI: So it depends on which set of resolver is you're asking. So for example, the authoritative resolvers like the root servers, those are open to anyone. The .com servers are open to anyone. The authoritative servers that answer that actually have the information and make it available, those are in general open to anyone.

Most ISPs resolvers however are only available to their customers, and that's largely for security reasons. If they're made available to everyone, they could get a DOS attack where lots of people from somewhere could start asking their inquiries and cause them issues.

There are also a number of organizations who have open recursive resolvers. So the same sort of resolvers that ISPs have that will look up the answer for you. And there are a number of companies who have open recursive ones. Google public DNS is one of them, the 8.8.8.8. Verisign has some. Open DNS is also a very well known open recursive resolver and they make their resolvers available globally and then have people watching them carefully to help mitigate DOS attacks, things like that.

WES HARDAKER: Good. So one addendum to that is just because the resolver is open, I mean, I just heard Warren said that there is three major resolvers out there that have known addresses, 8.8.8.8 being one, 8.8.4.4 for example being another one of Google's. That doesn't mean that your local infrastructure ISP company, government, whatever, is going to let you get there.

So there may be firewalls in place that prevents you from talking to them. Then sometimes there's corporate interest to track everything you do. And if you keep sending your requests somewhere else, that's not going to happen.

So we won't speak to the politics behind that but realize that you may not be able to get to some of these open resolvers based on where you're sending the query from and the local policies on the network there.

All right, next question, please.

UNIDENTIFIED MALE: And the question that I had is if you have a multi [inaudible] TLD like [co].in and in that situation, how does DNSSEC operate and how does the zone keys are managed in that kind of scenario.

WES HARDAKER: Good. You have an answer? Now, go ahead.

UNIDENTIFIED MALE: [Inaudible]

WES HARDAKER: No, I was going to ask [inaudible] again.

UNIDENTIFIED MALE: So you're saying if you have two level of zone within your TLD, how does that work?

UNIDENTIFIED MALE: Yeah, let's say I'm having [co].in, is the domain that we can get, we combine that xyz.[co].in. In that situation, how was DNSSEC operating because there are three levels of trust that I see and how is that trust transitioned?

UNIDENTIFIED MALE: So there's two ways. So you sign .in and then you sign [co].in as a separate entity and then you create a chain of trust between both. And then we can [co], you sign all the zones, the content with the [co] keys, that's one way of doing it.

And the other way, [inaudible] with .ta and then we have like a province, like om.ca and we treat that as a non-empty terminal. So we sign everything at the third level inside the zone just with the same key for .ta. So use .ta to sign the whole thing or you can use [co].ta to sign your own.

WES HARDAKER: So let me answer from a slightly different perspective and that one of the things that we left out of the skit, because it gets fairly complex, is that each one of the keys all the way down, you can have 25 levels.

As long as you start with the root key, there is a secure link between each parent and each child, meaning the root key knows the key for .in and can say, "Here's how you get there. Here's how you get the key. Here's the security link." And then .in would say, "Oh, well, I have children, [co] being one of them. Here is a secure link."

And so the resolver that is doing this chaining all the way down could get to a.b.c.e.whatever.in, as long as each level had the right link, it had that security link, then you can verifiably know that no answer anywhere in the entire chain had been modified.

Was that it?

UNIDENTIFIED MALE: Yeah.

WES HARDAKER: Okay. So do we have a question in front.

[SARATA]: Ghana fellow – I think my question was a bit answered with the last one but I wanted to find out the evil person that now we have signatures to reach, so we said he's not out of the system. I have a feeling that that evil person would now go into getting a signature himself and will be back. Is there a way of like just putting him off completely talking about the link in there?

WES HARDAKER: Warren.

WARREN KUMARI: So, yeah, one of the other things when we did the skit that we tried to, sometimes we do it, sometimes we don't. We try to make it easier and simpler. Is the bad person, Dr. Evil, whatever, where is the bad – okay, the bad person left because he's bad.

In some of the skits, he comes along with the signature. So the way that you can detect that that's a problem is .com talks to bigbank.com, Big Bank tells .com what the signature is. And so when the resolver, the ISP asks .com, "Where's www.bigbank.com," .com says, "Big Bank is over there and their signature looks like this."

So then when the ISP goes to bigbank.com, it knows what the signature should look like. When Dr. Evil comes along and tries to give the wrong answer, the ISP already knows what bigbank.com signature should look like and it can compare and see that it doesn't actually match.

WES HARDAKER: One other –

WARREN KUMARI: I'm not sure if that actually answered it or if it was clear, somewhat clear.

WES HARDAKER: I need to like one another important takeaway from this is again as I said in the beginning, DNSSEC protects the DNS. It doesn't protect you against other forms of attack that happen outside the DNS, one of which is social engineering. If somebody calls up .com and says, "I'm bigbank.com, I need to change my key," that might happen as a social engineering attack.

DNSSEC can't protect you from that. That's actually the registrar's job within the DNS world to make sure that they do verification of who you are when you are calling up a registrar to say, "I need to change some data."

So those types of attacks need to be addressed in a completely different realm than what DNSSEC is actually handing and there's actually been a lot of progress made there too. But that's sort of the outside of the scope of today's discussion about DNSSEC. Does that make sense?

[Inaudible], next.

UNIDENTIFIED MALE: Actually, I have two questions. Should I one-by-one or I just throw it out at the same time?

WES HARDAKER: We want you to throw them both out for now.

UNIDENTIFIED MALE: First question is that is the whole idea of the DNSSEC I think is great, but I also know that there are still some ISP they don't deploy at the DNSSEC. So what's the reason behind?

WES HARDAKER: Good question. [Inaudible].

UNIDENTIFIED MALE: The second question is that we also know that some recursive server provider, they don't actually go through the root and the [inaudible] because they get to the root zone from IANA and then just store locally. I think that Warren, you are one of the author for a [inaudible] back. So did that compromise to the channel [inaudible]?

WES HARDAKER: All right, so I'll take the first one. So in terms of – now, I'm blanking what your first question was. I shouldn't have had you do both one. Oh, why don't people deploy.

So there's a couple of reasons. One, they haven't heard about it. That's why we hold these sessions. We are still in the educational phase of the world. Not everybody knows about it.

The second one is that a lot of ISPs are understaffed and they simply – they may want to, they haven't had the time to sort of spin up new staff. The good news is a lot of the softwares, the DNS name server softwares, the resolver softwares are defaulting to turning it on. So that takes time for them to – when they get to deploying the next version, they may automatically get it and they may not even know that they turned it on because it actually is defaulting it on.

A lot of it just comes from users haven't asked. Have you called your ISP to say, "Hey, I notice that you're not protecting me. Would you mind trying to do that for me?" And if they get enough questions, maybe that will be motivation, maybe they will have an answer as to why they don't do it yet.

You'd have to go ask each individual why but I've heard all of these answers in the past, some being, "I don't have the manpower," some being, "Nobody has asked me to." Some people saying, "I don't understand it well enough to – I need to learn more first." So all of those are reasons that I've heard.

And then, Warren, you want to take on the second one.

WARREN KUMARI: So, yeah, your second question was a really good one. Just for some background, there's a document by the IETF which says that resolvers can simply keep a copy of the root zone inside themselves so that they don't need to keep going off and asking the root.

That doesn't break DNSSEC in any way because when those resolvers download a copy of the root zone, the root zone contains all of these signatures for all of the top-level domains inside it.

So when DNSSEC works, instead of it going off and asking the root, it looks inside itself and it has all of the signatures already or it has all of the hashes for the signatures technically, but it has the information for the signatures. And so it can still do the checking all the way down. It just doesn’t have to do a separate request to the root.

WES HARDAKER: The key that signs all the data in the root zone is owned by IANA and which is part of ICANN. And all of the servers that serve this, regardless of whether it’s a personal one or whether it’s the root servers, if you went to the Root Server Tutorial earlier today, all of them are serving the data.

And you can verify that they didn’t modify it. You can verify that your software didn’t modify it because the DNSSEC chain is started with that single key. And if you know that key and the one that Rick was talking about rolling, then you can verify what anybody is saying, where it came from.

And again, even if it went all the way around the room, no matter where you get that data from, DNSSEC doesn’t protect the transaction between you and me. It protects the transaction that tell you that ICANN – that nobody since ICANN published it has modified it that includes me, that includes you and anybody else that’s touched it.

So, yes, Warren.

WARREN KUMARI: And just a very short follow-up to that, the only reason that the root loopback documents exists is because of the protection, which Wes mentioned. The fact that DNSSEC signs all of the records means that you can take the data from anywhere. It doesn't matter at all even if you found out on a scrap of paper on the ground, as long as you take the data and validate it with the root key, it doesn't matter where it comes from. So you might as well have it inside your own resolver and not bother the root.

WES HARDAKER: Good question. We have another question.

UNIDENTIFIED MALE: ICANN fellow, I have a small question. My question is related to performance. How much overhead does this process causes? You are [inaudible] encryption for the root [inaudible] as well as all of the keys.

WES HARDAKER: It's a very good question. It's how much overhead, how much pain is there involved in actually doing the cryptographic stuff and how much slower is it? Does anybody have recent numbers? Rick? [Jack]?

[JACK]: So ISOC wrote a paper on this a while back and I think it was like five years ago. The numbers were back then around generating like 10% more load on the DNS but that was five years ago. Today, I would think is very minimal difference in performance.

WES HARDAKER: Warren.

WARREN KUMARI: So unfortunately, as with any technical question, it depends and it’s complex. So when you say load and performance, it depends on which part you ask. So it does require some more DNS lookups. There’s more DNS traffic, so there’s a bit more bandwidth.

But I think what you’re actually asking about was CPU or load on the resolver itself. So for doing the validation, I’ve seen some numbers published which is less than 1% CPU load for a normal sort of machine setup.

So that’s a very, very minimal or sort of not really noticeable CPU load for the validation. But there is additional lookups, there is additional roundtrips, but those get cached so they are also fairly minimal.

So the general answer is not enough that people really notice.

WES HARDAKER: All right, thank you. [Inaudible].

NASRAT KHALID: Hi, from Afghanistan. My question is about public DNSes. Why is it for free and why Google is trying to give us this favor?

WES HARDAKER: Warren.

NASRAT KHALID: And the next question is about the complications with – I mean, when you use that public DNS, you have complications in your own network and I've seen that all the time. And the latency, how much does that defer on resolving a hostname compared to a localized [inaudible] DNS? Thank you.

WARREN KUMARI: So, actually, I can probably answer both of those questions with the same answer. So the reason that Google provided this is because low latency is really, really important for users. If users' requests take a long time, they get bored and they don't use the Internet as much.

A lot of ISPs had very slow resolvers and so Google offered this because Google would like to make some money. Google offered this because that means users have a faster Internet experience, that means they use the Internet more and that means that they see ads more, which means Google makes money from that.

So Google doesn't charge for the service but it makes the Internet faster for users. It also makes sure that users get the correct information.

A number of years ago, there were some companies who were providing false answers. If you'd mistyped a name, they would send you somewhere else. And so Google is providing this to make sure users will get the right answers and so they would be happy and use the Internet more.

The latency to the Google public DNS should be better than your ISP. If it's not, then you should be using your ISP's resolver. If Google public DNS doesn't make your life better and faster, don't use it.

Did that answer both parts of the question?

WES HARDAKER: I think that was a good answer. Agree?

UNIDENTIFIED MALE: [Inaudible].

WES HARDAKER: He’s wondering about the data side of things.

UNIDENTIFIED MALE: [Inaudible] information because that was something Google is not [inaudible].

WES HARDAKER: So he’s wondering if Google uses the data that’s going on.

WARREN KUMARI: So Google’s public DNS privacy policy is written up on the web. It says what information we log and we log a small amount of information for troubleshooting, but then it gets anonymized and compressed. Most of it gets thrown away. But it’s all published online. I think it’s www.google.com/developers/public. I actually can’t remember. If you Google for “Google DNS privacy,” the information is all there and I will happily say that that is all true. If it wasn’t, it would end really badly. But I mean, I will personally say that’s actually what we do. I attest to that.

WES HARDAKER: So thank you, Warren, very much. I have a question up front.

[BETTINA]: I'm from Thailand Fellowship. A few years back, I have involved with trying to push the DNSSEC in Thailand. One of the things that's very hard for me is to defend for the budget because it's very complex, they don't understand basically. And then after I said because it's very technical, it's complex, they said, "Okay," so they had Google find it complex as well.

So basically, I cannot justify like how bad is the situation for the DNS hijack, I mean, comparing to the other cyber security attacks. Do you happen to have research or anything that help us for that? Thank you.

WES HARDAKER: That's a very, very good question. And that actually helps answer one of the previous questions of why don't people deploy validaters today? And some of it comes down to budget, so people's time always come down to there's a dollar cost associated with deploying anything new regardless of what it is. Rick, go ahead.

RICHARD LAMB: Well, I'm just going to speak because obviously, I have the DNSSEC religion, so I'm going to say positive things about it.

For many of us, it's not about protecting the DNS. What's cute about this? As you see, [inaudible] excite a bunch of these guys. Think about a system where we can now exchange key material, public key material, securely between anyone. You publish your public key, PGP key something. I want to send you an encrypted message end-to-end. I can do this now.

I use the DNS, I look at this key and I encrypt the message on my laptop, it goes to your laptop, you pull down my key or what have you, you have your copy of your key, you decrypt this.

Anyway, the idea, it become a global protected database that people can use to exchange information. This is what's really exciting and Vint Cerf even pointed this out like back in 2000. We all knew about it but in 2010 he goes, "You know, something much more is happening here. It's not just about [inaudible]."

So that's part of the answer. The other part of the answer is this infrastructure continues to grow, 90% of the TLDs have DNSSEC deployed, not the second level. Second level, it's only about 3%, it's really small. But that's opportunity to some of us.

So I know it sounds like sales.

WES HARDAKER: One of the other things that you can do is if you go Google for "DNS hijack," you will find that there is a whole lot of news articles about stories that actually have happened. I don't have a reference for you immediately off the top of my head of a catalog of them. But there's actually been so many articles published on things that have been taken over through DNS hijack specifically through cache poisoning or various other ways of poisoning DNS. The DNSSEC would absolutely trump.

So that might be information that you can go take back to anybody that is saying, "Why would I do this?" and say, "Look, there's actually been so many of these that it was so common that this is a no-duh."

Does it protect everything? No. I mean, it still goes back to what I said in the beginning, which is to protect everything, you have to sort of protect every layer of the Internet and the IETF is working very hard to protecting more and more layers [inaudible].

UNIDENTIFIED MALE: So one way we look at the DNSSEC, it's a platform for innovation. So the future, it's the Internet of Things. The Internet of Things is connecting everything to everything. That's IPv6. The Internet of Things obviously needs a lot of security around It, and that's DNSSEC, and that's called to the infrastructure.

That's how you sell it. You want the future, you want IPv6 and DNSSEC.

WES HARDAKER: All right, thank you very much. Is there anybody else that has not asked a question yet? We're trying to take new people first.

[GIGI]: Thank you. I'm from the US. And I'm just wondering if you can shed some light on how a new gTLD would go about implementing DNSSEC and how that would affect the registrants.

WES HARDAKER: Does somebody want to speak to the new gTLD model, Rick?

RICHARD LAMB: I'll speak to it. I really don't work much in the gTLD stuff. I try to stay away from it. So it's required for all new gTLDs to deploy DNSSEC. Most of the backend providers, when I say that, a lot of the gTLDs have other people running some of their infrastructure for them. They're all very well versed in DNSSEC.

How would they implement it? Well, you could take one of my courses. I offer a four, five-day course in various parts of the world. Unfortunately, no one in the US has asked for one yet.

Maybe you'll be the first. And at the end of that course, it's all hands-on. It shows you how to implement a full system, complete with key ceremonies and software and everything. You could do this all yourself.

If you don't want to do that and you don't want to use one of the existing backend providers, there are a few commercial solutions out there. Yeah, I'm not going to pick one or the other. And my personal view and because DNSSEC does require a certain amount of understanding, it's better to implement these things yourself at least to understand so that when something goes wrong, you will know what goes wrong.

I don't know if that answers your question.

WES HARDAKER: All right, thank you very much. We're almost out of time but we have time for one more question before the session runs out of time.

[AWAL]: NextGen, I'm thinking about for example if I try to resolve a domain along the path if [inaudible] Dr. Evil comes and try to do something as user, if DNSSEC is already along the way, it's working and as a user, can I get information or notification that someone tried to break that chain of trust as a user? Can I get the notification from the [inaudible] somewhere?

WES HARDAKER: Rick? Warren?

WARREN KUMARI: So as a normal user, probably not. It's very hard for you to discover that somebody tried to do it because your resolver, that resolver upstream from you, will just ignore the answers from him. It doesn't tell you that somebody has tried to lie.

If you're running your own resolver, you can look in the log files, the log files will say that it got a bad packet and it will just throw it away. So you basically should never see the bad answer from the attacker. But generally, you won't know that the attacker is there because you don't care about it. It doesn't make any difference to you.

I think that maybe I answered that very [inaudible] or the wrong question.

WES HARDAKER: I think that's a very good answer. And I will say that I actually wrote a GUI application that actually looks at log files and looks at traffic, and will give you a little pop up on your desktop if that happened. But you really have to be running the stuff locally on your laptop.

So how are we doing? We have one more time?

JULIE HEDLUND: Yeah, maybe one more.

WES HARDAKER: So one last question and then we'll have to close the session.

UNIDENTIFIED FEMALE: You answered my question [inaudible].

WES HARDAKER: Microphone please because we're streaming live.

UNIDENTIFIED FEMALE: You almost answered my question because he asked the question for end user how do we tell and you said you developed an app but it has to be installed locally. So my question was, do you foresee that it can be a plug-in for a browser so that we can just install on our browser whatever browser we use and then it will alert us?

WES HARDAKER: A very good question and so I'll take that one myself because I've actually written a plug-in for the browser that does just that. However, there's actually one – if you go to – there's one for both Chrome and Firefox that's actually a part of the plug-in repository that will catch most stuff because it doesn't look up every DNS lookup that you do, and there's a whole bunch of technical stuff that goes along with it of why it's not perfect.

We actually, at my previous company, wrote a complete rewrite of the DNS internals of Firefox to make it work. And so there's a browser out there called Bloodhound, which is the one that we wrote. That actually will do it at the user level.

But one thing to note is that in the same way that your ISP takes away all the pain of a whole bunch of other stuff like how to do routing, how to get packets to the right place, it takes away the pain of doing DNS lookup.

So in general, the end user doesn't need to know that the ISP protected them from a bad thing. The ISP is already doing that. They're already doing firewall and they're already doing routing, they're already doing DNS lookups for you, they're already doing managing the cables, they're already doing managing of boxes. And there's so much that goes on that the end user doesn't need to know about.

I would argue that a malicious packet that came in is probably one of the things that end user doesn't normally need to know about unless you're a geek like me.

So, with that, I would like to thank –

JULIE HEDLUND: There's one question in the chat. [Inaudible].

WES HARDAKER: Oh, one question in the chat, okay, since we haven't taken any of those.

JULIE HEDLUND: There's a question line from Paul. He says, "Have any TLD implemented DANE yet?"

WES HARDAKER: Excellent question. So the TLDs generally don't need DANE themselves. So DANE is a technology that we haven't talked about here, which is how to attach certificates gotten from a certificate authority or not to the DNS and to verify those through a separate chain.

And TLDs actually don't do that because TLDs don't usually have certificates very much. But end users do and actually, there's –

mail is where the DANE usage is really taking off the most and there's actually – if you go search for DANE and SMTP, you will find that there is a documents written and there's a whole website that catalogs I think the last check, the number was 300,000 different domains that were using it. Many of which came from some ISPs in Germany that they really wanted to stand up and make it happen.

Did you have something to add, Warren?

WARREN KUMARI: Yeah, I just want to mention following up from that. Rick said that DNSSEC protects the DNS but it also creates a platform for you to be able to do other things. DANE is one of the things that allows you to do.

And, yeah, as Wes was saying, it's now being used a lot in e-mail [inaudible], which is the US national standards body recently published document basically sort of suggesting this. And the German organization for information security or something like that also has made DANE e-mail a sort of strong suggestion.

WES HARDAKER: All right, thank you very much and let's thank our panel for helping us out this evening.

All right, and if you have any questions, remember that the DNSSEC deployment website is actually one that has been worked on for a long time and has a whole bunch of answer to the questions like this. All right, thank you very much.

JULIE HEDLUND: And let's thank Wes, the one and only Wes.

[END OF TRANSCRIPTION]