RAW FILE

ASIA‑PACIFIC REGIONAL INTERNET GOVERNANCE FORUM

NEW DELHI, INDIA

13:30

04 AUGUST 2014

TAPAS 3

SESSION: DNSSEC DEPLOYMENT

Services Provided By:

Caption First, Inc.

P.O. Box 3066

Monument, CO 80132

1 877 825 5234

+001 719 481 9835

www.captionfirst.com

\*\*\*

This text is being provided in a rough draft Format. Communication Access Realtime Translation (CART) or captioning are provided in order to facilitate communication accessibility and may not be a totally verbatim record of the proceedings.

\*\*\*

(Please stand by for the APrIGF Conference 2014 Session: DNSSEC Deployment.)

>> DON HOLLANDER: So we'll wait about five minutes, I'm told. So when the big hand is on the eight. And this is the session on DNSSEC, a layman's review, for nontechnical managers. Are there any geeks in the room? Anybody want to be admit to being a geek? You may be in the wrong place, except our speakers are generally geeks.

>> DON HOLLANDER: So, welcome. My name is Don Hollander and I'm the general manager of APTLD. This is about DNSSEC from a layman's perspective. It's a small group and we have three presenters, two with slides and one short presenter as in not very tall and also not very long. So I have organized this session because people asked to look at DNSSEC and what it's about and why it's important and what are the missing bits in terms of getting DNSSEC fully deployed throughout the Internet ecosystem.

So from my perspective, I want to know the answers to that. So I have gone out to the community and found a couple of experts. So what we'll hear today is about different components and it's very relaxed. It's a small group. Please, if you have questions, ask them. Interrupt politely. You are welcome to come sit near the front. There are desks you can put your computers on and we'll go ahead and get started.

For people who are participating remotely, there's a young man here who is going to be monitoring the remote participation channel, and he will also ask questions on your behalf. We are just trying to get all the bits and pieces. Technical bits. We have the slides up on the screen. Champika is from ICANN and he will do the rest of his introduction himself and we will get started. Champika, are we ready now? We will start with this microphone and then shift to a different microphone.

>> CHAMPIKA WIJAYATUNGA: Okay. Good afternoon, all. Let me introduce myself, I'm Champika and I represent ICANN. Probably already you know about ICANN. So I won't go too much details into that. Now, as Don mentioned earlier, the objective of this session is to ‑‑ all right. Okay. Yeah. So as Don mentioned to you earlier, the objective of this session is to discuss about the DNSSEC operation from layman's terms. So we would assume that, you know, you are not really not technical people because we are going to go into very basics and to give you some kind of an understanding about this mechanism called DNSSEC and how it is working and what are the things that you have to be aware of if you are to go and implement DNSSEC, right?

Now, first of all, before we really understand DNSSEC, we need to understand about DNS. I mean, what is the purpose of DNS and what we are trying to achieve using DNS. So simply DNS ‑‑ I mean, before I actually start this presentation, I would also like to thank Rick Lamb, who is also our ICANN's DNSSEC senior program manager who has also provided some of these slides as well. So we would like to acknowledge him. So as I mentioned to you before, when we talk about DNS basics, what you need to understand is about what is the purpose of DNS? Now, as we know, DNS is a mechanism where we can translate one type of object to another type of object or you can even call it as a resource. So we translate one type of resource to another type of resource. So when I type resource, this is like IP, and domain names and so on. So you can see that, you know, if you can see the data communications that happen in the Internet, it's all happening in the network layer, right? So we have multiple layers of communications. So we have this layer called network layer. That's where we use the IP addresses. So simply if I put it in this way, you see, if you want to get connected to Internet using any device, it could be a computer.

It could be any mobile phone, any device, what you need to do is you need to have an IP address, because we need to have some identification to connect to the network. Now, so we use an IP address to define the uniqueness or define the identity of this device. So the thing is, we use IP addresses like IPV4 addresses which are pretty long where we have 32‑bit address space and we use IPv6 addresses which are even more complicated where we have 128 bit address space, right? But from a layman's point of view, these are all addresses. The thing is, these are addresses we cannot remember easily. This is why we need to have some mechanism in the first place to, as humans to have a mechanism where we can try to remember these addresses in a better way. So this is what the DNS system is doing. Basically it will map these addresses to a name, right? So that's the whole purpose of DNS.

Now, you can also call it as kind of a distributed database, if you consider the whole DNS named space. This is a distributed database where we have lots of these names put according to a certain organized structure, right? So this structure is where you have kind of a three hierarchical structure where we have a root on the top, and then the responsibility of each named space is delegated to multiple other parties. So we are talking about a tree hierarchy. And this is where all what we call top level domains comes in or TLDs. So we nowadays we have lots of TLDs, for example, like dot coms or dot net, dot edu, we have things called generic top level domains, gTLD or ccTLD. I would assume that you are familiar with those, for example, in India, we have.in. That's the country code and then dot‑coms and dot nets, and so on. There are plenty of those. So basically in this tree hierarchy, the top of the tree is what we call the root. So the root structure. This named space, if you consider the named space, we also have to identify where do we keep this name space, right? It's a kind of database, in a technical term, you can call it as a file. So we have this database, file, or even a name space, and we need to have server to keep the name space. These are name server. So we talk about name space and then we talked about named servers. So named servers keep this name space. Now the next question is: What is the objective? What is the purpose of this named service? Okay. They keep the named space, but for what? To answer their clients. Because we have servers to answer their clients, or in DNS terminology, we call those resources. All right? So we talked about three important things, the named space, then we have named servers and then we have clients or what we call resources. So things in DNS revolve around these things. When we established DNS in the early days, the security was not thought of very much. The main objective was to get the data from one place to another place and then to make sure that we can access those hosts and servers. Right?

So, for example, even when we try to browse a website, or even when we tried to send an email, we used this DNS system, okay? So that's ‑‑ that's how we used the DNS system. Now let's look in a simple way, say you are going to browse a certain website. If you are going to browse a certain website, how is this going to happen? So we have this revolver, which I mentioned before. Resolve is a Klein. For example, if you are using a browser, so that's your client. And would you type www.ICANN.alt. When you do this request, what you are trying to find out is the IP address. The IP address of the ICANN web server. Right? So this request will go to a named server which we mentioned before. Remember? I talked about three things. The second one is the named server. So this request will go into a named server. Now, the objective of the named server is basically in this case, is to find this IP address and return that to the client, which ‑‑ which that question was asked.

So this named server, this ‑‑ we call it as a recursive name server because this is kind of a recursive process, which means it will go from one server to another server and so on, and then try to find the answer, basically. Now, we have ‑‑ remember earlier also I mentioned to you, we have put all of these name spaces into a proper structure where we have the root on the top. This is what we call the root zone. Zone is an administrative space, right? Zone is an administrative space where we have the mappings of these delegations. So earlier, I told you that we are delegating. So when this query goes to a root server, root server will not, of course, give the whole answer of the IP address of ICANN.ald server. Instead, it will return back with an answer to recursive named server with a reference. It will say, I have a delegation to, in this example,.org, because we are looking for the web server of ICANN.org, right? So it will refer to you to the named servers of dot org. And then the recursive server will go and talk to the dot org named server.

In this generic example, I have put gTLD. GTLD is a generic top level domain. So this could be dot‑com, dot net and so on. If it's a country code, it will go into a server like IN, for example. And then the gTLD server will return back with the IP addresses of the named servers of next level. So the next level is in our example which is the ICANN server. So actually, there's a mistake over there. I have APNIC, but it should be ICANN server. And then ICANN server will reply back, right? ICANN server will reply back with the IP addresses of that web server. Okay?

So this is how the query happens and then this IP address will be returned back to the client. The IP address will be returned back to the client and then there's one more thing that happens which is what we call the caching because once this IP address is returned back to the customer or the client, basically, in this case, that IP address will be kept in the cache or in the memory of that named server. So that in case ‑‑ if the same question will be asked by another client, if you are trying to try to browse the same URL, for example, then the named server can give you a quick answer because the named server doesn't have to go into all of these root servers and our gTLDs and so on.

Now, talking about the root servers, it is not only just one server. It is scattered around the world. Right? So we have primarily 13 of these root servers but we have lots of instances of those, in addition to those. So if you take, for example, the first root server is named as the A root and the B root and so on, all the way up to the 13. And then there are lots of instances of these root servers around the world. In fact, in India, we have a few of these root servers located. If you do a query in here from India in this room. Most likely, you will go to one of the root servers. You don't have to go too far to resolve this. So this is a very basic DNS query and how it works. You need to understand this process for us to actually talk about DNS security in the first place, right?

Okay. Now I also talked about the cache. That is the TTL. The TTL means time to leave value, which has been kept as the caching value. All right?

Okay. So let's now try to analyze, what are the flows or the data flows involved in this process. Now, first of all, I told you that we have to have a zone, right? We have to have a zone what is a zone? I told you before, it's an administrative space. If I'm going to refer, right, if I'm going to delegate something to someone, I should know how to get to that point, right? Say, if I'm dot‑com, if I'm dot‑com, I'm delegating Google.com or something dot‑com to another party, I should know how to get to that party and that should inbound my administrative ‑‑ that should be in my administrative space or what I call the zone. So this delegation is represented or mapped in a file called zone file, right? So that's what we call a zone file. So so we have to have the zone file where we have all the zone data. We have all the zone data and we have that in this zone file. This zone file, let's say all the DNS administrators, they will configure their named server and then they will create a zone file for that particular zone that you are interested. Say, you are going to go and get a new ‑‑ your own zone, right? So say myname.com. So you need to create a zone file for yourself, which is my name.com and you do all of your delegations inside that. Now we can also update the zone file dynamically as well. So that means even from a remote location, you can do that dynamically, this is another area of DNS where we call it dynamic DNS. Now, the things are getting a little bit complicated when you think about the redundancy, because if I'm running a named server, in this case, I'm referring to another name, what we call an auto rotator server. That means this named server is authorized to give answers for a particular zone. Earlier, we talked about the recursive server and this is an authoritative server. So it gives authorizations for that. I can run a named server which will give answers for my name.com which is what we call an authoritative named server for myname.com. If I want to have some redundancy for that, what I would do is I can run what we call a secondary named server. So we talked about primary named servers which is actually the server I'm running and then as a redundant server, I can run another secondary named server. That's still an authoritative named server. And these secondary named servers can be of multiple servers. It needs to be two or three, multiples and then I can also ask many other parties. I don't need to be myself. I could be myname.com, but I can ask somebody else to be a secondary. We have masters which is the primary and secondaries which we call slaves. When the master has updated all of its information, right, things like, you know, www.records, all of that is inside this zone. Now this zone has to be transferred to the secondary so that consistent answers can be given to any query coming any of those named servers.

Now, for better redundancy, I can have my primaries and secondaries in a different place. If ‑‑ there are some organizations trying to ‑‑ they would try to put your primaries and secondary only in one subnetwork, which is very bad, because in case ‑‑ if the subnetwork is attacked all of your primaries and secondaries can go down. And then some organizations would tend to put all of your primaries and secondaries in one building, but this is also not very good because the building can get taught in fire, floods, all of those natural disasters. If that is the case, both your primary, secondaries and so on will go down.

And soU. try to distribute those secondary servers. It can be multiple countries, right? You can have your primary in one country and a secondary in another country and so on. So if you can put it as much as, you know, in different places, that's even much better for your redundancy. So now we have these ‑‑ if you have multiple of these named servers, we have to make sure the content, the zone files, the zone data, they are consistent across all of those named servers we have, right?

So that's where we have to actually look into because we have to have a process where we will transfer our zone from one server or the primary to a secondary server. And then what happens? Remember I told you about the recursive server? There are thousand, tens of thousands of recursive named servers who will come and talk to those primaries and secondaries to find the zone data. Right? So, for example, you know, my machine here could be talking to ‑‑ I want to find what is www.Google.com. And then the query has to go to Google.com's primarily and secondary. If I want to go to yahoo.com, primary and secondary named servers and so on. So it should go to all of those different places. And so for that, we have all of this recursive named server. Now recursive servers they will contact the masters or the slaves. The masters or the slaves, the primaries or the secondaries. The primary could be in one country and the secondary are in another country. There are mechanisms to make sure that we will access the closest one possible. Then we have our clients who are doing all of their DNS queries. They are trying to browse something and so on. So you can see there is kind of a number of data flows across these functions, what we have discussed so far. Now, the question is: How secure you are. How do you make sure that no one tamper with this whole system? How do you make sure that you are safe, your DNS infrastructure is safe. Do you see some place where someone can take some advantage over you? Do you think someone can attack you no some particular place? So these are the things that we have to look into, when you think about DNS security.

If you do a guess where you think you are safe. If you take a typical DNS structure, where you have these processes one, two, three, four, five, none of those maces are safe enough because if someone want to take advantage over your DNS infrastructure, they can. So, for example, number one, look at number one, someone can change your contents inside your zone file. So you might be having a record called www.myname.com has got the IP address x.x .1, but if that IP address has changed, say to y.y .1, that means your zone data has been tampered with. That means your primary named server will give wrong answers to those caching servers. That means the cachers will get polluted or poisoned.

That means the wrong answers will go into the client. So they will ultimately visit wrong websites, right? They will not go to a correct website. They will go and visit the wrong website. So zone files can be interrupted. What else? Think about the master and the slave. We have primary and secondary. So they have to trust each other. When the secondary goes to primary to get the zone, how do the secondary know that this is really my correct primary? They could be talking to some wrong party. Maybe someone is impersonating, they are pretending to be the primary. This is also possible. In the same way, someone can pretend to be the secondary. So I'm going to send my zone file to a secondary, my server, but maybe someone else trying to pretend ‑‑ pretending to be the secondary server. So these are possible. And then what else?

We have caching servers, or caching forwarders, the recursive servers. So they can also contain wrong information in the caches. The caches can get polluted. Wrong information can be injected into those caches and so on, right? So this is why it's very important for us to protect our DNS infrastructures. If you don't protect any of those places can be attacked, can be basically taken advantage over us, right? So this is why the whole purpose, why we need to have some DNS security mechanisms so that we can try to project our DNS infrastructures.

Okay. So this is something like any other security mechanism we have in the world now. So when you go through, you know, various ‑‑ various places, or even when you browse through, you must have met so many security mechanisms. Maybe you might be getting some parsing, maybe you have some devices where you will have, you know, checking, two‑way passwords kinds of things and then you may be checking on various smart card functions, all of those things, or passports, identity cards, all of these are ways that we can try to introduce some sort of security into our usual system where we will go one step ahead of securing our place, or environment.

So DNSSEC is just another of these things and so that we are trying to actually go some step ahead and trying to protect our DNS infrastructures so that we can make sure that we are visiting correct websites, we are visiting correct places. We are not talking to the wrong parties and so on.

Okay. Now, where will it fit in? Basically when you think about DNSSEC. Earlier I talked about DNS mechanisms. I showed you the whole structure and where we have to protect and so on. So the main theory behind this is when you try to go and implement DNSSEC there are some things you need to look into. Now, this is where the keys are all kind of assigning process will come in. Right? Now, the whole purpose of this thing is for us to avoid those man in the middle of attack, which I mentioned to you before, in case someone has tampered with our zone, we should be able to identify that. In case if someone has injected something wrong into our zone we should be able to know that our zone file has been lost. These are the things we are trying to actually achieve using this DNSSEC. Now DNS stands for DNS security extensions. So what we are trying to do here is to protect our DNS infrastructures in a cryptographical way. So you may have heard about PKI or public key cryptography as well, where we use keys. In DNSSEC, we are not trying to encrypt the thing. We are not trying to encrypt because we are talking about DNS public data, but what our objective is to make sure that we talk to the right parties, we go into the right places and we can ensure that the integrated of our data is not lost. It's intact. That's the whole purpose of that. Simply, we are trying to protect your data. We are trying to protect our zone data, okay?

So in this ‑‑ in fact, probably, you may have also recall in the recent times, there has been various of these criminal attacks and so on, which came through DNS vulnerabilities and DNS changes was one that created a lot of loss and this was mainly because of the things I mentioned to you earlier, the configurations were changed. So the integrity has been lost and so the client would not know that. They would go and talk to some wrong DNS servers or wrong web servers, right, different places. So if you try to put this whole thing as a phone book, right, DNS's system is something like a phone directory or a phone book. You can see in this diagram, we have the DNS revolver and the DNS server which I mentioned to you, before and then we have a web server which we are trying to access. So what happens, in this case, the URL is something called www.majorbank.se. So you want to try to access the major bank website. You would type in that URL and that would go into a DNS revolver, right, and then that will basically go into the DNS system and as we discussed, it will find out the IP address of the web server so that you can then visit the appropriate web server. That's what normally happens in a usual DNS transaction.

Now, once you get into the web page, obviously, now this is a bank bang and you might give all of those credentials, the log‑in credentials and then you try to basically access your data. So these are kind of your typical case. Now, as I mentioned to you earlier, during this transaction, we will go through that DNS hierarchy that we highlighted before. It will go into the root. It will go into the next level, in this case,.SE and then the second record and then further down, it will go into the majorbank.se and then the website. It travels through this hierarchy.

Now, where he also have, as I said to you before, we also have this caching responses for faster service, efficiency, so what happens is that when the ‑‑ when the answer is cached in your memory, you don't have to have that DNS revolver, it doesn't need to go into the DNS server, it can be given straightaway, given back to the client. So the client can directly access the website, right? So there's one step lower. So we can save some time there, right?

So it's the same function we do.

But when it comes to, say, someone who wants to really try to, you know, do some bad thing and trying to, you know, attack the system, instead ‑‑ you know, before you get an answer from the DNS system, the correct party, the correct authoritative server, someone can pretend to be the correct party and they can try to inject in this wrong answer to the revolver, to recursive server, which will then in turn return the wrong address, the wrong IP address of a fake web server. It could be a bank web server, very fake. It looked like the real site but it's a fake site. They can return that IP address and that will go into the client so the client will end up, in visiting that wrong web page, it looks the same. So they might end up in giving the same password, the same credentials and so they will capture that and put into a database and then the user might think, oh, okay. I just mistyped my password and then they can offer ‑‑ they can close the transaction, which will go back to the normal one, right? So this is a kind of very typical scenario where cache poisoning attacks can happen and the user can be fooled, all kind of, they will be taken to a wrong place.

Yes, so ‑‑ and also sometimes more than this ‑‑ now, this is again, the same sort of answer with the caching. So if that wrong answer gets into a cache, to a recursive server. That means recursive server will keep this always, in the memory, in the phonebook. Any time a customer comes in asking for this bank website, the recursive server will start giving up all of these wrong answers, fake answers. So this is another problem, obviously we are poisoning our caches so that caches give all of these wrong answers. All right?

So now, what we are trying to do with DNSSEC here, is we are trying to protect this DNS data so that if some response comes from an attacker, we should be able to identify that it is a fake response. Right? So we should be able to know, you know, the DNS recursive server should be able to understand it can not validate this wrong response coming from this fake server, right?

So this is what we are trying to achieve using DNSSEC, I mean, one of the possibilities we are trying to actually get out of DNSSEC.

I mean, how we can do this, in other words, what we are trying to do is we are trying to sign our DNS data, right? I'm using a term signing here. What do we mean by signing? Signing is, it's a kind of process where we incorporate kind of digital signature into the data in our zone file, right? So this is what we are trying to do. So to do this, we need to actually have what we call keys. Right? Keys. So we are going to have, in fact, key pairs. So what one key pair is what we call a private key and the other key pair is what we call a public key. Now, first of all, let's analyze, why do you really need to have these keys or what is the purpose of these public key? And what is the purpose of this private key? So, for example, you know, if I try to talk in ‑‑ usually in a real world example, when you want to go to a bank, to cash a check, if you want to go to a cash a check, you would go to the bank and then they would actually check your identity, right? They would check your identity. They would check your signature, and they will try to validate you. And if it is all fine, they can allow you to get your money by end cashing that check. It's a similar situation here. So we have got our data. We have got all of those resource mappings inside the zone and then if I'm a recursive server and if I get some kind a mapping say, www.ICANN.org has got this IP address, I should be able to check that, whether this is the correct information, but I can validate it using a significant, a digital signature. So with the keys, what you are trying to do is we have two keys, one is a private key and one is in a public key. We have generated these keys when we deal with the DNSSEC. So once we use these keys, we use this private key. That is something your own key, right, so you will keep it within yourself. You will put ‑‑ you won't give it to anyone, it's your private key. So you will use this private key, and then you will send it through some kind of a function, right, which will generate a signature. Right?

So first of all, you generate these keys, private key and public key, they will be generated using some type of algorithm or mechanism. Then you use the private key and send it through a function where it generates a significant. Now we have a significant. Now the private key has sort of helped us to create a signature out of that. So signature can be shared across anyone. So I can give my signature to you, and you can validate me based on that signature. If you see my signature, you should know this is Cham's signature, right? So you should be able to validate it. How can you validate it? That is the question. That is why we have another key for that, which is our public key. So we have two keys, as I said to you before. I have my private key, and then I have my public key. So I signed my data using my private key and then I generated a signature and then that signature can be validated using my public key. So I have to publish my public key. Otherwise you wouldn't know that. I can publish my public key and then you can take my signature, and validate whether it's really coming from me. So if we can do this, the servers, you know, those recursive servers and so on, the clients, you know, if they can get this information, they know before they visit a certain website, before you get into a certain name, you know exactly whether it's a valid name or not, whether it's a fake name, whether someone has spoofed it and so on. That's the whole purpose of this exercise. All right. So that's the previous example which I was showing, you know, what ‑‑ you know, accessing the web page of the bank and then giving all of your account details, right?

So with the validated records, if you can validate, then we know exactly we are going to the right place and getting the correct information from accessing the correct URL and if there is some bogus thing, you will drop that. That's the advantage of this whole function. Right?

Now, this is the simple technology behind what we are trying to do within DNSSEC and then there's lots of these DNS security hijacks. There are only so many of these threats and attacks and so on that has happened. Now, sometimes some of you might wonder, okay, we already use the HTTPS and other certificates. Why is it different to the DNSSEC and so on?

Now, the only thing is that with the SSL, that's the transport layer in the layers. There are authorities, what we call certificate authorities, who will actually certify your signatures, the certificates. Now, the thing is there are so many of these certificate authorities actually available in SSL type mechanisms and there are thousands of those. You may not know exactly which one to trust and which one not to trust. In the past, there have been lots of issues related no this as well. So this is why actually, if you can use DNS to do this validation to have this function, then we can rely on the place that we are visiting. All right?

So from a, you know, cyber security point, you know, even if you have to do a business case for this, if you have to go and actually justify why you need DNSSEC, these are the main things that you have to bring it in, because DNS is one of the very critical infrastructures we have and especially we deal with every day. We deal with emails and we deal browsing and all of these devices and so on and we do that all the time. So if you do not have a proper securey this in your infrastructures, that can be quite ‑‑ the infrastructure can be quite vulnerable to all. These attacks and so on, right?

Now, you know, talking about some status updates and so on there, has been a lot of interest from various bodies various parties and governments to implement DNSSEC. We have Sweden, Brazil, Netherlands who adopted early in the ccTLDs and they are zoned and so on. Nowadays we see there are lots of other operators, service providers and so on. They also moving to this DNSSEC mechanism as well and then there are some government mandates, for example, US government, they have a mandate which actually happened a few years ago as well. So in this way, there are lots of interests being generated among various communities, various forums, various stakeholders and so on to implement DNSSEC, right?

You can see in the recent times, these are some statistics, actually, in the recent times, many of these zones, many of the TLDs and so on, they have implemented DNSSEC. Actually, if you wanted to find some more more information, statistically, I think the ISOC website, they have got more statistics based on ccTLDs and so on. So these are, again, you know, a few statistics related to deployment stuff based on last month, the 29th of July and so on, out of 654 TLDs, we have ‑‑ you know, DNSSEC is enabled in 462 in this. That's a good percentage, 70% or so. Now, the other thing is that roots ‑‑ remember, when you remember the tree hierarchy which we mentioned earlier. In the tree hierarchy, root is a top and root zone is sign. Root zone is already sign, which means when you try to actually validate something, this trust has to be propagated to different levels of this hierarchy. So as we know, the roost is on the top. So the top ‑‑ the root has to be, you know, DNSSEC enabled otherwise it will be a little bit of a problem because we are talking about a big hierarchy, starting all the way from root. You see we have the root and the top level domains and the next level domains and so on. We have to make sure that the trust has been propagated throughout this hierarchy. This is why root signing is very, very important. So signing of root has to be of important value. So which has happened already. So root has signed. And ‑‑ and also most of the TLDs. They have signed as well. And then there are ‑‑ there are various other ‑‑ you know, in case if you are looking at signing options and so on, there are various third party solutions also, there are various other projects going on, working group, like DANE is one of the working group. DANE stands for DNS based authentication for named entities. Earlier, I mentioned SSL. SSL, so we are trying to actually have this protocol, where we can incorporate the DNS system to support SSL based answers, right, replies basically. So that's watt DANE is. So these are all developments happening actually in the DNSSEC area. This is not something imaginary. These things are happening. The number of TLDs, they have already signed. Root has signed. We are moving forward. It's very important. We need to think about this option.

Yes, the sad news is not many second level domains, they have signed, right? So as you see, the presentation, it's very small percentage of second level they have signed. So here's why it's very important for us to actually get this message across, and people understand this technology, people understand this mechanism and then get them ‑‑ get their zones signed. At least try to do it in a test environment, production, you know, if you can't do it in a production environment, start from a test environment and start from there. Okay? So when you ask about, okay, some of the organizations they do various problems. They don't have enough capable in the organization. They don't have enough technical staff in the organization to implement these. Some people are worried with the performance issues, in case if you implement DNSSEC. The performance of the DNS infrastructure can be affected. Some people are worried about the increase of the zone sizes. There are various replies given, but the thing is that once ‑‑ once you are familiar with this thing, it will come in place, right. Main thing, as I said, it's not too complicated if you try to do step by step. So the whole purpose is try to give some thought. Try to actually, you know, find more information about this. Feel free to talk to us. We do conduct lots of training courses, capacity building activities around the region or so in terms of, you know, making sure the DNSSEC message is actually passed across the community.

So here as I said earlier, we are trying to actually generate this trust. That's the whole purpose of DNSSEC. Given the time constraints also I wouldn't go too much detail into this whole diagram, but the whole idea is that we have a long hierarchy in DNS, all the way to root and so on and we have to propagate the trust all the way from the top to the bottom. That's the whole purpose. Yes, so what you can do, for the companies, think about signing your domain names and you already have that and in turn the validation on the DNS revolvers. For users, we have plug‑ins. That means you have the clients. You can have these plug‑ins in your browsers. So when you do a DNS query, then they can validate whether they are installed or not. And also as I said, take the advantage of all of our educational activities and so on that you will then be ‑‑ get some more information about DNSSEC.

Yes, I think, you know, that's the main message we wanted to actually give you, especially during this type of a session and then we have ‑‑ okay. I think I talked about ‑‑ okay. A little bit from the trust side. Earlier when I mentioned to you about the root, sometimes I also told you that in SSL, we have many certificate authorities. We have thousands of certificate authorities. We can not trust those. But in DNSSEC, when we propagate this trust all the way to the top, we are talking about, you know ‑‑ you are talking about child/parent type of relationship. If you are my zone, mydomain.com. And the parent of dot‑com is root. So that's the type of relationship we are talking about. I told you earlier that root signing is very important and how do we make sure that it is trustable and our root keys are safe and all of those things. Some people may be wondering about those things as well. So these are examples about the infrastructure related to the root keys and they are kept in a pretty safe places and also they are being shared in a very safe environment, and also there are community members, selected community members from all part of the worlds who are gathering into this root signing, key signing ceremonies. So they actually do it in in a very streamlined way, right?

So here's something that just on the trust side, and basically, we need to address those challenges today and make sure our DNS infrastructure is safe. That will help for the whole DNS. We are talking about the tree.

If we can get more people into the DNSSEC signing then actually the DNS query, what we do in general can be safe and secure. So I think that's the message that I wanted to pass across. It's a very quick overview of DNS security and we do run workshops which can go into multiples of days, actually, if you try to do this in a technical environment and so on. So if you have some staff engineers who are willing to actually do some more configurations and more technical details and so on, feel free to be in touch with us but I will now pass it to Don and also my colleague from Afilias who will also share some more information related to this. And then we can take some questions and answered. Okay.

>> JITLENDER KUMA: Thank you, Champika. My name is Jitlender Kuma and I will from Afilias. We will talk about today as Champika has already given you a very good overview of what DNSSEC or how can we secure the DNSSEC while implementing the DNS security extensions. So we ‑‑

Okay. Thank you and sorry for the technical glitch out there. As I was saying, as Champika has given us a good overview of what DNS is and how we can utilize the DNS security extensions to prevent for any damage or any attack on the DNS. We'll go a little bit into the DNSSEC, as well as how we go from the registering of the DNS ‑‑ sorry, registering of the domain names and how we can go in as a registrant or a domain affects us and how we can go ahead with this one. So first, I would like to take a moment to, you know, talk about Afilias, the company I come from. Afilias is best known for their domain name registry services. So we provide the back end domain name registry services to our customers, as well as we provide back end domain name registry services to some of the top level domains or gTLDs as we know them. So dot info is there and dot org. Of the new top level domains are coming up. We also provide back end registry support to some of the ccTLDs and we are very proud to be associated with India' top level domain country name. Now as Champika has covered in detail of what DNSSEC is. So I will go really quickly over. So DNSSEC basically it's a set of security extensions, which is added to the existing DNS protocol. It was designed by the Internet engendering task force which works on developing standards and protocols for the internet, right? What essentially DNSSEC provides is the authentication of the source of the information in a DNS response. Now, just a little bit to exmain what it is, as Champika was talking about the diagram the user types into the browser, something like www.Google.com and the browser then sends the information to the local DNS provider which could be the ISP's provider, right? While going through all of this there are multiple instances are, you know, in between, there are levels of where your query passes through and if somebody knows the protocol very well, they can alter the response to give you the false data and take you to a website that's not really authentic, right? And towards the end we also heard about, you know, some people might argue we use the SSL certificate for security of our website. So why should I go for DNSSEC, right?

Primarily SSL certificates are majorly used for encrypting the data that has been transferred from the user of the website, and they don't really provide the authentication of the domain name. They can only say, yes, in the browser says this is www.Google.com and it matches the common game of your SSL certificate, that's about it. They are not going to check anything else, right? So it's not very secure, you know, way of authenticating whether this domain name that you are visiting, whether it is, you know, properly authenticated or not. The integrity of the information in a DNS response is also provided by DNSSEC, which means the query you have sent or your browser has sent to the DNS server when the response comes back, DNSSEC ensures that the response you are getting is the same response which was sent by the server and there's no tempering of data in between. Authenticated denial of existence, which means if the domain does not exist, if you are getting a response that the domain does not exist, that response would be authenticated and legitimate. It's not like somebody is redirecting you to basically spoofing the data in between and saying the domain does not exist. Wherein you have the data, this basically happens in the cases where the domain already exists but somebody is attacking that domain with multiple queries that the servers are so busy with the customers would want to get to that domain name. In that case if you get a denial of existence response, DNSSEC ensures that's a proper response, and not a spoofed response.

However, what DNSSEC does not provide, is the confidentiality, the access control list, as well as other means of differentiating between the inquiries. Right? And protection against DOS attacks which means the denial of service attacks. That's not handled by the DNSSEC. That has to be handled by maybe providing ‑‑ putting on the ‑‑ some of the rules on your ‑‑ some of the rules on your devices, the network devices which can prevent excessive query handling, right?

There are two key principles which are involved in the DNSSEC. The first one is the signing, wherein you sign your existing domain name with the DNSSEC using the cryptographic signatures and the second part is validating that signs data, right? So these are the two principles which are part of the DNSSEC implementation.

We were talking about the chain of trust. Champika did mention the chain of trust. How the chain works, I think the diagram does not appear clear on the screen, but at the top, we have the root. Right? And then we have the zone or the TLD zone wherein your domain reside and then we have the zone file of your domain name, right? And then if you have any third level delegation within your organization. So at the bottom we have that. So that's how the chain completes. Each level, each parent level contains the authentication data for their child zone. So that any request which comes for the authentication, it starts from the top, goes down to the, you know, the smallest level and each level basically authenticates the response received from the parent is correct or not. So I will just ask a quick question here, how many of you have registered doe main names for ‑‑ domain names for yourselves? Okay. You must have filled the form with your registrar where you registered your domain name and provide them information such as your contact information and sometimes you also ‑‑ if your DNS provider is different from your domain registrar, then you have to provide them with at least two named servers as well so the domain can reside on the Internet. So once your doe ‑‑ domain is signed these are the new resource records which get added with the signed data. If your domain name is not signed, the usual resource records would be such as the area code, which is for IP address and email exchanges and whatnot, however when you sign your domain name, these four records get added to your domain name data in the zone file.

Okay. Let's talk about the signing of the domain name, affiliates. We have been signing TLDs, even before when the root zone was signed. We are also responsible for key material for signing process, such as the publication of the signs keys or the keys that have been signed from the registrants once you signed your domain name..IN registry, it was one of the early adopter of the DNSSEC. I believe we signed the.IN in 2008. It was made available to the registrar and that they can opt for signing their zone in November 2009. So if anybody over here who has a dot IN domain, they can take the leverage of this facility which is provided by IN registry. They can go ahead and soon their domain and make sure that their domains are secure. Right? Some of the registrars which you may be aware of, are Go Daddy and names beyond India whohave been the early adopters and made it available to their registrants in India and globally.

After your domain is signed, comes the part of validating wherein, you know, we provide that hard hitting response when a query comes for a zone name. So if your domain name is signed, we would be basically ‑‑ your revolver will be validating whether the response that has been received by ‑‑ you know, from the server, whether it is ‑‑ whether it is authenticated or not, or if it is correct or not, right? We also provide the DS record that enables the validation of the signed domained in the TLDs that we are hosting. However here, registrars are responsible for ensuring that registry has the public information it needs for publication of the record. So as a domain owner or as a domain registrant. When you go to your registrar and we want to sign with the DNSSEC. They need to have the two keys, they would be basically providing the registry with this DS record which contains information about your public that would be scored into the registry zone data which will be utilized for authenticating or ensuring that, you know, whenever somebody comes for your domain name, the query your zone data. So we provide ‑‑ we basically validate before providing that information over to the requester.

Now, where does the gap come into the system? Right? So the public information needed for the DS record is managed by the DS hosting provider. So when you register a domain name or when you basically go for domain name hosting, there can be two different providers there. You may be registered a domain name with one of the regular Starr but your DNS server could be different. You may want to host the domain within the organization as well but you may have somebody else who is hosting your zone, right? So everything is fine, as long as your registrar is your hosting provider because that single entity would be managing your domain name, as well as the domain, or the hosting party. When it comes in picture, there is a constant need of interaction between your registrar and your DNS hosting provider: Right now it's a manual copy and paste, where you copy whether you are a registrar and you provide your data to the DNS hosting. You provide your zone data to the hosting provider and then the hosting provider will give you the named servers which are hosting your ‑‑ which hold the record for your domain name and you provide that back to your registrar and they would provide that into the inclusion of the TLD zone. So right now it's a copy and paste mechanism that's going on. When your domain is already signed, and, you know, in case you have to change your registrar for whatever reason, it could be bad service being received or anything. So you want to change your registrar from A to B, and your domain is already signed, you need to take utmost care while doing those domain transfers there's a very fair possibility that it would not resolve for a certain amount of time. So we will see how basically it works. The key objectives that we will talk about here are to ensure that a valid zone transition happens from the old service provider to the new service provider. While you are doing the transfer. If you are a gTLD or TLD dot compliance with the gTLD rules that have been established, the problem is when the DNS services are linked with the registration process which means, you know, while you are registering the domain name. If you are opting for DNS services as well from the same provider, then it should be good at that point. Once you have your domain name and it's already signed with DNSSEC and you are transferring own the DNSSEC which means only the hosting part to different provider, what would happen? All you need to do is generate new keys from the new DNS service providers where you will be migrating your domain name and add it to the zone, existing zone. You sign with both the providers and publish your zone at both providers, which means that you go to the new DNS provider. You sign your zone and get to your keys and give to the existing regular Starrs where ‑‑ registrars. Once the registrar publishes the DS records, you just need to do the changes into the named servers at your registrar so the domain would be automatically turned over. You are cord fating with both the providers and your registrar is helping you with that part. After your transported over no your DNS services to the new provider, you can easily discontinue the services at the old provider. This is the scenario where the transfer of domain name comes when you have the bundled DNS services with your domain name. So the registrant, which is yourself contact the new registrar where you to move your new name. It transfers on behalf of the registrar. So there are certain procedures and processes that the registrar may be following wherein you have to provide the authentication, you know. So you have to provide certain authentication over there that I want to transfer my domain name over there. You have to provide that written email sometimes that, yes, that is fine. The old registrar would receive the request after, that, you snow the new registrar Sens that request to the registry. The old registrar would be notified about that. There is a general grace period of five days if the transfer is not approved by the new ‑‑ by the old registrar, the registry waits for days and during those five days, their domain may go dark at that time, right? So once the domain is transported over to the new registrar, the new registrar would then change the named servers because you will be taking the hosting services at the new registrar and they will be hosting your zone from the beginning and then change your named servers to one of the named servers of their own. At that time, during those five period ‑‑ five days grace period, the domain would ‑‑ depending upon the time to deliver the TTL value of that particular zone, it the domain may go dark. What is needed here to ensure that we have the seamless transition or the doe main does not get affected due to that, we should be able to deploy the DNSo the DNSSEC in advance the registration so they can prepare before the transfer actually happens and add key signing keys where the new registrar would create those public and private keys and give it to the registrar and they can go to the old regular star and say add these keys as well, in addition to what I already have. Export the new key from the new registrar and you can import the new keys from the old registrar into the registry. The DNS services at the old registrar should continue to work as a test, because of the old keys which are already working. After the DS gets changed, you can go back to your old registrar and they can remove the old keys which are no longer in use and the new keys would pick up from there. The ideal process for doing that would be the registrar contacts the new registrar. They apply the new sign zone. They export the keys and export the new DS records and they add the old registrar. Import the new keys which have been received from the old registrar, from the new registrar and much the name server record as well as the DS record at the registry, wherein now the registrar has the change and we wait for the transfer to complete. And then it pushes it to the registry and you can discontinue your DNS at the old registrar. So this coordinated process would ensure that the domain resolves at all the time and is secured with the DNSSEC, okay. Some of the references that you can go through, Internet Society has a page where you have all the resources available and you can watch videos and go through their tutorials or the white pages out there to enhance your knowledge to see how the processes work for signing your domain name to resolving them on the Internet.

So with that, I will, you know, pass it back to Don to see if there are any questions and then we will determine whether we can help you with any of those questions. Don, over to you.

>> DON HOLLANDER: So thank you very much, Jitlender and Champika. The bit I was going to talk about was already covered. I will take one minute to reemphasize the chain of this ecosystem. So all the people involved in supporting DNSSEC, you have the root zone operator, which is ICANN and then the top level domain operators, those are the ccTLDs or the gTLDs and we heard that there's about ‑‑ of the 650 some odd top level domains in the root now, more than half are signed. That's because I can required the new gTLDs to be DNSSEC enabled.

Then you have the registrars who have to be able to support DNSSEC and Jitlender explained how it can be. If you want to have a simple environment where it's doing the same serving, very easy. Very straightfor. They take care of it. But not all registrars support DNSSEC because there's effort involved.

And then you have your ISPs, who also have to validate the DNSSEC operator and there are two ISPs involved. It's the ISP that's providing the hosting service and there's the ISP that's being used by the end user of your website or email or application. And they need to be able to validate, to check to do that checking that DNSSEC is signing and that you are actually getting to the place that you are working. And that's an area that's still missing in a lot of places. And then the final area that is missing, generally, is the end user experience. And Champika told us that there are a number of plug‑ins that people have created that you can add to your Firefox or Internet Explorer or your Safari or what have you and it will tell you, oh, there's a problem here. But that is ‑‑ requires and extra effort by the end user to configure to their browsers an you have to figure out what that means. And then you've got issues with mail validation and then the applications that people are using more on their phone that actually never see a ‑‑ you never see a domain name at all. You just fire up Facebook or eBay or your banking application on your telephone. So from a layman's perspective, much of the technical work has been done. It can be very easy. But there's these couple of bits still missing. So the TLDs generally on to it. The root definitely on to it. The registrars, some of them support it. The ISPs, some of them support it. Not too many, actually. But some. And some big ones. And then the end user experience. So that's the issues ‑‑ the DNSSEC is clearly going to be very important, very useful. Champika made it very clear that if you are using a bank, you would want your bank to take as much security precautions as they can to make sure that you are going to the bank you want and not somebody else's virtual bank.

So that's just the overview. Focus on the ISPs, Champika told us about and the end user system.

So with that, I want to thank the presenters or doing that. And it's now time for afternoon tea, but I'm quite happy to stay for a few minutes if people have questions. Are there any questions?

(Off microphone comment).

>> DON HOLLANDER: Champika, you can probably hear better and you can probably answer the question because I'm not ‑‑

(Off microphone comment).

So the question was, do antivirus packages do the validation and Champika's answer is probably. Yes, sir?

>> AUDIENCE MEMBER: Hello, is it sufficient to check whether our bank hog or online connection websites having HTTPS, as a layman as a end user, I want to make sure if I'm doing a transaction, so generally we have HTTP and HTTPS. So if we see that it's showing HTTPS, now it's okay. Is it really sufficient against ‑‑

>> DON HOLLANDER: Can I try to answer that? So HTTPS encrypts the traffic between your PC and the end where you are going to. It doesn't confirm that where you are going to is actually where you want to go to. So with HTTPS, if people somehow incept the traffic, they will not be able to to understand it, but it could be that it's going to the wrong place. Is that the right answer?

Phew! Let's ask another question.

>> AUDIENCE MEMBER: So as an end user, what are the ways to ensure that we are going to the right place, we are accessing the right place? What should be in our laptop? What are the tools that should inbound our laptop?

>> DON HOLLANDER: Yep. So there are some plug‑ins and Champika he will let me know if I'm right. There are some plug‑ins that you can download and put into your Internet explorer or Firefox or whatever, but you also need to make sure that ISP is validating and if they are not, whether you have the plug‑ins or not won't make any difference. And there are places on the Internet that you can go that will tell you if the ISP you are using is validating DNSSEC or not. So if it's not, then you call up your ISP and say, could you validate the DNSSEC and the help desk operator that you get will say, hmm? If enough people call up, they will figure it out.

>> AUDIENCE MEMBER: My question is why some website with the DNSSEC certificate is not working in all web browser?

>> DON HOLLANDER: Because I have old ears, and speak ‑‑ and just slowly.

>> AUDIENCE MEMBER: Why is some website, have the DNSSEC certificate is not working in all browser?

>> DON HOLLANDER: So the website is ‑‑ the DNSSEC is not a certificate‑based validation. It's in the root process. So does that make sense? What you might be seeing is if you are going to a website, that is signed, it's part of a TLD that's signed. So everything is signed and kosher. And when you go to the website, and you have ‑‑ and your ISP is validating, when you go to the website, if there's a problem with the key the result you will probably get is it can not find the website, which is a good solution. So it doesn't say, oh, there's a problem. It just says you don't want to go where you are trying to go because it's not associator. ‑‑ kosher.

>> AUDIENCE MEMBER: But sometimes it's working on Internet Explorer 8, but it's not working in Internet Explorer 11.

>> DON HOLLANDER: Oh, I don't know.

>> JITLENDER KUMA: It's not resolving? (Off microphone comment).

>> Maybe the browser doesn't have the layest version.

>> DON HOLLANDER: Okay. So the question is: Why doesn't it work? And the answer is: Different because we think you are talking about certificates as opposed to DNSSEC signing. So a newer browser will have a different range of certificate authorities than an older browser maybe. But it's not a DNSSEC issue. It's a certificate issue.

Even one last question and it's time for tea and the presenters will be around for tea. So just ask them questions.

>> AUDIENCE MEMBER: Is there any rule regarding the redemption? Redemion period rule?

>> DON HOLLANDER: Oh, I will let ‑‑

(off microphone comment.)

>> AUDIENCE MEMBER: I know that, but some resellers, it's for gTLD, that the redemption period is ‑‑ they say you have 40 days and then the different resellers have 30 days. In general is there any rule?

>> DON HOLLANDER: Let me see if I understand the question. Is there any consistent policy on redemption grace periods that applies to all TLDs? So the answer is no.

>> AUDIENCE MEMBER: With the reseller?

>> DON HOLLANDER: So the reseller will just have to use the policies of the registry.

>> I would just add one thing to this answer. So the 30 days what I told you is a registry policy, however, 30 days is given to the registrar or reseller in that case, and if the registrar wishes to, you know, delete the doe Pape ‑‑ sorry, they can delete the domain and get their money back, right? So this is the reason for keeping ‑‑ because once your domain gets auto renewed or transferred or anything, you know, any financial transaction has taken place, and if the end customer is not paying for that financial transaction, registrar has the right and the time to delete that domain and get their money back, which they have spent on behalf of the registrar. So that's when the grace period comes over there.

Okay? And the 30 days grace period is something which is for ‑‑ you know, it's the registry policy and I wouldn't ‑‑ I wouldn't know if any reseller or anyone would would say the redemption grace period is of 14 or 15 days, because the more ‑‑ or the longer this period is, it's beneficial for them, because they have a customer that has the right to go back and get their domain activated again.

>> DON HOLLANDER: But the time for the registry grace period is up to each registry. I think 30 days is fairly common. So with that, thank you very much for coming and for staying, actually. And for the nice comments that I got as I worked ‑‑ walked around the room, and please enjoy afternoon tea. Thank you very much.

(End of session)