Julie Hedlund: Thank you everybody, this is the DNSSEC Work – DNS Session for Beginners, and I'm going to go ahead and turn it over to Simon McCalla.

Simon McCalla: Thanks, Julie. Good afternoon everybody. Thank you very, very much for coming to – and filling the room for what is the third – the third go at DNSSEC for Everybody. I'm very aware it's been a long day with some quite big announcements, so thank you for coming.

I hope that we're going to entertain you in the next hour. It's going to be a light-hearted session. I'm hoping that you'll learn something about DNSSEC, if you've never – if you've never understood DNSSEC before, or you've worried, it's too complicated, we're going to try and keep things very simple, going to try and avoid too many technical terms, hopefully have a bit of fun as we go this afternoon, and hopefully you'll leave knowing a little bit more about DNSSEC.

There will be a section at the end of this to ask questions. So we'll have a sort of round table, if anyone would like to come and just talk about what you've seen, and what you've heard today.

As Julie said, I'm Simon McCalla. I'm from Nominet who run the dot UK zone. I'm very, very blessed to have a panel of DNSSEC luminaries and experts here with us. We have Matt Larson from VeriSign, a very large part of helping create the DNS Standard;

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.

Russ Mundy from Cobham and Sparta who has tirelessly campaigned for DNSSEC roll-outs, and we have Norm Richie who is part of ISC and of course the World's most popular name server with bind and make sure that DNSSEC is integrated into that, and we have Roy Aarons from Nominet who is part of creating the standard for DNSSEC. So you couldn't ask for a better, more qualified panel today. So thank you all for coming along.

So without any further ado, briefly we're going to try and rattle through in an hour these topics. So hopefully we're going to cover everything from the basic concepts through to some real world examples of DNSSEC implementation.

What I want start with; I want to start with a story. I think many of you will think that DNSSEC was invented by this panel over here and done so at about ten years ago, and blah, blah, blah, et cetera. But that's all lies. Actually, DNSSEC was invented 50,000 years ago in 5,000 B.C. That's not 50,000 years, my math is poor today.

So actually DNSSEC was invented by these people, this is Ugwina, she's a cave woman, and she lives on the edge of the Grand Canyon, and she's here, she's dressed up for the occasion, and there's a reason for that.

On the other side of the Grand Canyon is her boyfriend, and he's called Og, and he lives in a cave. And they look at each other across the Grand Canyon. It's a long, long way down to the bottom, and it's a long way around. And so poor Ugwina and Og don't really kind of get together that much, it's pretty difficult for them.

So one day when she's finally having a visit, and they're having a chat, they're sitting around the fire, and they notice there's smoke coming out of the fire, and they say ah, hang on a minute, there might be something in this; and by the time she's gone back to the other side, they're using smoke signals to talk to each other. And they're very happily sending smoke signals across the Grand Canyon.

But unfortunately, a new mischievous caveman, a man called Kaminsky moves in. And he's a bit of a naughty guy and he quite fancies Ugwina too. So he starts sending her smoke signals. And now the poor girl is pretty confused. She doesn't know who's trying to chat her up.

So off she goes, sets off down the Grand Canyon and tries to sort out the mess that's clearly happening. So Ugwina and Og consult with the elders, and there's a caveman called Diffie, and he's sitting there and he says, ah, I think I might have an idea; I can sort your problem out for you. So he jumps up in a dash, he runs into the back of Og's cave – in Og's cave, sorry. And they're all what is he up to, what is he up to.

And at the back of Og's cave, he finds a pile of very strange blue-colored sand, and he can only find this sand in Og's cave. And he's like I know what I'm going to do next. He grabs a handful, and he runs out and he throws it onto the fire. And the smoke turns this amazing blue color, and off he goes with a great whoosh.

And so now Ugwina and Og can happily chat again, she knows I'm only going to listen to the blue smoke, and poor old Kaminsky, well, he's left scratching his head on how he's going to chat her up next.

Now, fundamentally, that's it for DNSSEC. I mean somebody tries to tell you it's really complicated, all that DNSSEC is about is about making sure that your smoke signals are received and not somebody else's smoke signals. So if you remember one thing from today, and one thing only, remember that blue smoke. When somebody says to you what does DNSSEC do, it allows my DNS queries to become blue and make sure that they reach the target that they're intended to reach. And that's it. It's as simple as that.

Now we're going to talk a bit more in detail now about exactly how we made that blue smoke and what happens next. But I hope that's an image that you'll keep in your mind. I'm going to hand over to Matt, who is going to take you through DNSSEC in a bit more detail.

Matt Larson: Thanks Simon. Good afternoon. So this is DNSSEC as fast as I can do it. I'll try to talk slowly. And in addition to DNSSEC, I also wanted to start with a little bit of really high level DNS intro material, just so everybody is on the same page, and we're all using the same terminology.

So let's start with the DNS stuff. This is the name space. This is the structure of the distributed database that makes up DNS. And everybody is familiar with the root and top level domains and second level domains. You maybe haven't seen them represented like this. Computer scientists call this an inverted tree, because the root is at the top, and the branches grow downward.

And when we talk about domain names, we're simply giving a name to a particular node; we call it, in the name space. So here we have www.cnn.com. You can see that we formed the domain name by working our way up the tree and reading off all the labels with dots in between.

And the idea behind DNS, is that we can associate pieces of information with individual nodes in the name space, and that means that those domain names for those nodes are associated with those pieces of information.

So for example we could associate an IP address with www.cnn.com, and that's really the main reason we have DNS is this name to IP address resolution. But there's other information in there as well. There can be other pieces of information associated with any of these nodes in the name space.

We talk about domains. The definition of domain is really straight forward; it's just a particular node in the name space, and then everything below it. So here for example, I've circled the HP.com domain. So we have HP.com at the top there of the circled area, and then everything below it encompasses the HP.com domain.

Now in contrast to the term domain, we have the term zone. And this is really the reason that we have these slides up front is to explain the concept of zone, because that's really important to understanding DNSSEC.

And the idea behind zones is that they reflect administrative boundaries. This whole distributed database, the whole name space is too large to any single organization to administer and that's not the point anyway. DNS was designed so that we could have distributed administration. And that's where this term zone comes in.

The zone boundaries reflect how we cut up the name space so that one organization can manage one part, another organization can manager another one and so on. So we start at the top of the name space with the root zone, and then the zones are created by delegation.

So a particular zone that we call parent delegates to a child zone, and then that child zone in another delegation context, it could be a parent and have children and so on. And this can proceed down the tree.

So the root zone for example delegates to all the TLD zones. So the root zone is relatively small. If we look at a TLD zone like dot com, you know there's a hundred million plus domain names underneath dot com. So the dot com zone is huge.

But the dot com zone then doesn't include information like say you know in the HP.com zone. And I've also pointed out here you can see the difference between the HP.com domain and how that is sliced up into zones. This is actually how the HP.com domain is administered. There is an HP.com zone, and then at the level below HP, individual portions of HP have their own sub-domains, which are zones.

So what that means is you could have the folks at HP labs, that's HPL instead of HP.com. They can administer their own zone, their portion of the name space, and they can do that completely independently of the folks at corp.HP.com, the corporate IT people. So that's the idea behind zones, that we can distribute administration, have different organizations manage different parts of the name space.

So zones are also important because this maps to name servers. And name servers are how you tell the world what the information in your zone is. Name servers effectively publish the information in your zone. So for example, there's a set of servers for the root zone, the root zone is very important, as we're going to see in a moment, because it's where resolution starts. You know there's a set of servers for TLD zones, here's an example of the set for dot com. And then for cnn.com here, we have a set of – set of name servers.

So now, let's look at the resolution process knowing that we have a set of name servers that know about the root zone, a set of name servers that know about the dot com zone, and finally a set of name servers that know about the cnn.com zone. So what we have is some internet user at the lower right, sitting at their monolith shaped computer, this is what happens when you have an engineer create the slides, and not a graphic designer person.

So some internet user is sitting at a computer, they type in their browser, www.cnn.com to get to the CNN website. So their browser communicates with a portion of their computer's operating system called the stub resolver, and that's the operating system's DNS client. It's job is to take requests from applications who need DNS information and turn them into DNS queries that can be sent to name servers.

So the stub resolver sends a query to something called a recursive name server, and this is the part of DNS that does all the heavy lifting. As we'll see it knows how to look things up. Wherever you have a bunch of stub resolvers, you'll have a recursive name server. So for example in you know an organization of even modest size with a bunch of PCs or MACs or whatever, likely the IT department will be running a recursive name server. ISPs have to run recursive name servers for all their broadband customers. So there are a lot of recursive name servers on the internet, millions in fact.

But let's go back to this example. So the stub resolver sends a query to the recursive name server saying, hey, I need you to look up the address of www.cnn.com. So to make this example more interesting, we'll assume that this recursive name server has just been powered up, and it doesn't know anything. That's not completely true, though, because in order to start resolution, you have to start somewhere, you have to know something. So the one thing that recursive name server does know is the names and IP addresses of the root name servers. That can't be discovered by a recursive name server, it has to be configured, though, you have tell it literally the names and IP address.

So the recursive name server does know those, and it can go to a root server and say, hey, can – you know looking for the address of www.cnn.com. Now the root server knows nothing about that particular information. It can't answer the desired query. So what it's going to do is do its best job to answer, it's going to give the best information it has, which in this case would be a referral, we call it to the dot com servers. So the one thing that the root does know is well of the query asked of me, I do know, I can talk about dot com. So I'll send a referral to the dot com name servers.

So the recursive name server follows that referral it now knows all dot com name servers, it can pick one of them, send it a query; ask the same thing again about www.cnn.com, and the dot com servers, again, they don't know the address for www.cnn.com, but they do know about cnn.com, there's a delegation to cnn.com. So they know the names of those name servers. So they can return a referral to cnn.com.

You can see where this is going. The recursive name server now sends the same query, a third time, and this time to the cnn.com name servers, it gets a response. Now, I should note that all of these responses have been cached, that will speed up resolution in the future for somebody else who asks the same query, or something else in cnn.com or even something else dot com. That recursive name server will use all of the information that it has cached to short circuit the process.

So the recursive name server returns the result to the internet user, the web browser then can connect to cnn.com and get the web page displayed. I do know enough PowerPoint animation to do that.

All right, so that's how DNS works in a nutshell as fast as I can do it. So armed with that, those concepts and those terms let's go into a brief overview of how DNSSEC works.

So I just want to see how I'm on time, okay. So DNS, DNS Security, let's talk about DNS security, well that's easy, because before DNSSEC, DNS really had no security, it was designed in a friendlier time when there just wasn't thought given when the protocol was designed 25 plus years ago to secure – from a security perspective.

DNS also uses one packet for a query and one packet for a response in the majority of interactions. And what this means is, and I should also say that DNS uses source IPs for authentication and as we know relying on source IPs isn't a real good mechanism for authentication, because they can be spoofed, IP addresses can be spoofed. Someone can forge a source IP address, and if you're using only a single packet to contain an entire query, and an entire response, it's very easy to spoof a response.

Now, over the years in the absence of something like DNSSEC, people who are resolvers, or in this context, we're talking about recursive name servers, you know they got clever and tried to do the best they could make recursive name servers paranoid, so that they would only believe what were hopefully legitimate answers, but you can only do so much by being clever. We really needed something else to solve this problem.

And the answer is DNSSEC. And DNSSEC uses public key cryptography and digital signatures, and we get two things with DNSSEC. We get what's called data origin authentication. We know based on DNSSEC that a given answer came from a given zone.

So you would know that it came from, say, you know the dot com zone or the VeriSign dot com zone, or whatever. You know which zone it comes from, it can be proved cryptographically. And you also get data integrity, which means you know that the information hasn't been altered since it was signed. So you know that it hasn't been manipulated in any way. The information you're seeing is the same as it was when the zone owner signed it.

So what this does is offers protection against spoofing, because DNS data is cryptographically signed, you can verify that signature and have some measure of confidence that you're looking at authentic data.

Now what doesn't DNSSEC do? Well, it doesn't provide any confidentiality. In other words, there's no encryption here, we're not keeping anything private, because after all the data in DNS is public, so we're only signing data, we're not encrypting data.

It also doesn't address any attacks against the name server itself. So this doesn't protect against you know problems with implementation, you know like packet of data or buffer overruns. It also doesn't protect against denial service attacks. So it's – DNS security is strictly focused on one aspect, and that is preventing spoofing by authenticating data.

So when you're doing public key cryptography you have the public key and a private key, what's called a key pair. And in DNSSEC, each zone, this is why as I said a moment ago, it was so important to have a good understanding of what a zone is, each zone has a key pair, a public key and a private key.

So that private key were going to be used for signing. So only the zone's owner should know about the private key. So that private key has to be stored securely. Nobody knows about it except the zone's owner, because they're going to use to sign all the data in their zone.

And the way public key cryptography works, is you sign with the private key, and then you need the public key to do the verification. And so the public key then needs to be published everywhere, so that somebody can verify your data. So you keep your private key private, you sign the data and then someone obtains your public key, and they look at that digital signature, they verify it with the public key, and if it checks out they know that you, with your private key actually made the digital signature.

So the zone's public key goes to a new DNS type of record, we call it, called a DNS key record. So DNSSEC public keys are published in this DNS key record.

All right, so as I've said, the zone's private key then signs on every piece of data in the zone and the digital – the resulting digital signature goes in another new type of record for DNSSEC, called an RRSIG. RRSIG is short for Resource Record Signature. So the public key goes in the new record type, and then the resulting signatures, made with the private key, also get published in this new RRSIG record.

Now, let's talk about how you actually validate data. Let's say that the cnn.com zone, they want to do DNSSEC, so they create one of these key pairs, a public key and a private key; and they use their private key to sign all their data, and then they publish their public key, and they say well here's our signed data, here's our public key, you can just use our public key to verify the data. Well, how do you trust that public key? How do you know that that really is cnn.com's public key?

You don't necessarily have any reason to know, and that's where this concept of a chain of trust comes in. The idea is that you can start with something you do trust, and if a key you do trust signs let's say another key, that signs another key, that signs another key, that finally signs a piece of data that you want to verify, you can work your way backward doing those validations until you get to a key that you trust. And so now you have a chain of trust.

You start from the key that you trust, when it signs another key, now you trust that key, when it signs another key, now you trust that key; and when that finally signs a piece of data, well then you know that the – the signature is valid and you trust the signature and you believe the data is authentic.

So we have a chain of trust. How does that work? You may be familiar with how, we call it X509 or SSL certificates where in the – most people use those certificates, they encounter them in the web world, you know using SSLs for ecommerce type stuff.

And the way digital certificates work is you can shop around for what's called a certificate authority and you can have different third parties attest to your public key's legitimacy. So I don't want to get into too much detail here, but basically what a digital certificate is, it's a statement that says in the case of SSL, it says this web server has this public key, and then it's digitally signed. And the digital signature is made by a certificate authority.

And with this model, you can go to different certificate authorities, and your web browser, most web browsers trust many different certificate authorities, so you can choose. Somebody might choose one certificate authority, somebody might choose another. And because the web browser knows about both of them, both digital certificates will be accepted as valid.

You don't have that option in DNSSEC. There is not this concept of certificates, and further, you can't have different third parties vouch for your public key. You can only have your parent in the

DNS hierarchy vouch for you. So this chain of trust is rigid. It starts at the root, and it works its way down, just as delegation starts at the root and works its way down, likewise, this chain of trust starts at the root. The root signs for TLDs, which signs for second level domains and so on.

Now, I wish I didn't have to go into this little bit of complexity here, but in order to give you the – you know the complete picture here, it's necessary to talk about how they are actually two keys per zone in most circumstances. You have basically one key that sends another key to sense data in the zone. We have a key-signing key, that signs what we call a zone-signing key, and then that signs all the data in the zone.

And the reason we have this is that this allows you to keep those two keys separate, the key-sending key as you will see has to go to your parent, so in most cases that means an interaction with the registrar, whereas the zone-signing key all that needs to do you can keep that completely to yourself, you don't have to tell anybody about it, it signs the data in your zone.

And what this means is you can change the key-signing key without changing the zone-signing key and the zone-signing key without changing the key-signing key. So this is an operational – it's for operational expediency. It makes it possible to change the different keys. It makes it – the real reason is that it makes it possible to change the key you sign your zone with, without having to talk to a registrar.

Sorry, I couldn't just keep my finger on the next button, I keep hitting previous. All right, so this brings us to the final record type we're going to talk about which is, we'll call it delegation signer record.

I said that a parent vouches for a child's keys. So that's the key-signing key in most cases, and the way the parent does this is the child sends not the key to the parent, but a cryptographic cache of the key, so basically a smaller representation of the key, that can stand in for the key, and then the parent signs that, and so that cache of the key, that representation of the key goes in what we call a DS or a delegation signer record, that's in the parent. And that gets a digital signature.

So the next thing and near the end here, that we're going to talk about is the concept of a trust anchor. You have to trust somebody and as I said earlier, you have a known starting point that you trust, and from there you can build a chain of trust. So you have to have these trusted keys, called trust anchors and chains of trust will start at all your trust anchors. So it's a key that you trust implicitly and this is analogous to the list of certificate authorities that your web browser knows about and trusts implicitly.

So let me wind up with an example here that kind of hopefully will tie this all together. You know this is a really – a fairly arcane topic and I'm going through it very quickly, but I think that this diagram will hopefully pull it all together.

So here we have – I've got the root zone, the dot com zone, and the cnn.com zone, and you can see I've highlighted the address for www.cnn.com, so let's talk about how that address for www.cnn.com; let's talk about how that piece of data could be cryptographically signed and someone could validate that piece of data and know that they were getting the legitimate version of www.cnn.com.

So recall that zones have – zones have keys. So I'm showing the public key, because that's what gets published. But remember for every public key that we publish, there is a private key kept private that is actually used for the signing process. The public key is only used to do the validation, the verification. So cnn.com has a key, and also remember, well that's what I just said, public and private.

So that key is used to sign the IP address of www.cnn.com. Now, I'm also showing that that one key for cnn.com is really two keys, it's the key-signing key and the zone-signing key, but to avoid cluttering the picture here, we left that out.

So the key for cnn.com, we make a representation of it using this hashing function and that goes in that DS record, and then that gets cryptographically signed by dot com's key. We do the same thing with dot com. Dot com's key we have a representation of it in the root zone, and it's signed with the root zone key. It's really signed with the root zone's zone-signing key, and the root zone's key-signing key signs the zone-signing key.

So that the root zone's key-signing key, that could be a trust anchor in this case. And many people, who are doing DNSSEC, do just that. They configure the root zone's key as a trust anchor. So in other words, they implicitly trust this key.

ICANN manages that key, the root zone's key-signing key, once every quarter, it signs the current zone-signing key, because we change the zone-signing key for the root once a quarter, VeriSign, my employer, hold the zone-signing key, and we use it to sign the data in the root zone.

So if you trust that root zone key-signing key, you can build a chain of trust all the way back to trust the signature on www.cnn.com. And let's look at how that might work. So if you trust the root zone's key-signing key, it signs the root zone's zone-signing key. So now you trust the root zone's zone-signing key. Well, it signs the DS record for com.

Now recall that the DS record is not dot com's KSK, it's a representation of dot com's KSK. So because you trust the signature on the DS record, you trust what it signs, which is a representation of the dot com key-signing key, so now you trust the dot com key-signing key. Well that signs the dot com, the zone-signing key. So now you trust that key. That signs the cnn.com DS records, so now you trust the cnn.com KSK. That signs the cnn.com's ZSK, and that finally signs the address for www.cnn.com.

So hopefully that helps bring it all together. This all happens in the context of DNS resolution. So first, that recursive name server looks up www.cnn.com and it gets the data back as usual, and then DNSSEC while it also gets the digital signature, and then it has to do this process that I've just described to validate that digital signature.

And it potentially starts all the way with the key for the root, and does this – builds this chain of trust and again, because it trusts that key for the root, it's configured as a trust anchor, it's going to trust no matter what, that's what a trust anchor means, because it trusts that it can build this chain of trust, and finally trust the information it got for www.cnn.com.

So that at a high level and kind of quickly is how DNSSEC works. So now I'm going to hand it over to Norm.

Norm Richie: That was a lot, you can see there is a lot of jargon that's involved in DNSSEC and there is – although that's a higher level, you can imagine how detailed the thing is.

For a bit of fun now, and as penance for creating DNSSEC, and all the terminology, we're going to do a play. So we're going to act out – show you how the transactions actually work, not quite as fast as actual servers run, it's a bit slower, we're going to use a piece of paper to pass around the transactions, and we're going to turn people into servers. Okay.

So what we have here is our authoritative servers, so we have root, com of course from VeriSign and Big Bank. And here we have ISP and I am Joe User, and the play comes in four acts, and we're going to do some online banking, as everybody probably does, we'll go through the process then we'll – we'll tell you the next acts when we get to them. It's a very high budget play.

So Act One. Joe User does his online banking, so here I am, I'm at my terminal; www.bigbank.com.

Thank you Joe User, I'm the resolver at your local ISP. I don't have this information in my cache, and therefore I need to resolve it. I do have the root address in my cache, and therefore I can go ask the root for www.bigbank.com.

Hey, Mr. ISP, I'm Senior Noel, good to see you again. I'm afraid I don't know the address for www.bigbank.com, but I do know where dot com is, and you can find him at 1.1.1.1.

Perfect, thank you. Hello, dot com, I'm looking for the address of www.bigbank.com.

Well, I don't know the address of www.bigbank.com, but I can tell you that bigbank.com's name server is at 2.2.2.2.

Thank you dot com. Hello bigbank.com, I'm looking for the address for www.bigbank.com.

Well, I'm glad you came to visit, and I do have the address for www.bigbank.com; it is at 2.2.2.3.

Perfect, thank you. I now have the address for www.bigbank.com and Joe User; you want to go to 2.2.2.3 for your banking.

Thank you.

So there we go, now I can go my online banking. So what happened there? This is slow motion version of how a DNS query works today without DNSSEC. So as a user, I know very little, I rely on my ISP to get me all the information, the ISP happily talks to the authoritative servers, and as we explained before, they have a hierarchy to them. So that is blistering fast, I know.

Okay, Act Two. Act Two is Joe User get scammed.

So we're just going to repeat this again and show what can happen in today's environment where we do not have DNSSEC, and it will show the importance of DNSSEC.

Doing my banking again, a lot of bills to pay, okay, www.bigbank.com.

Hello, Joe User, again I don't have that in my cache, I forgot the last session and so I have to ask the root for more information. Hello root, I'm looking for the address of www.bigbank.com.

Hey, Mr. ISP, good to see you again, and so shortly, I'm afraid I don't know the address to www.bigbank.com, but I do know where dot com is, and he can be found at 1.1.1.1.

Thank you root server. I'm now going to the address of the com server at 1.1.1.1. Hello, dot com, I'm looking for the address of www.bigbank.com.

Well, I don't know that address, but I can tell you that the bigbank.com's name server is at 2.2.2.2.

Perfect, I'm now going to the bigbank.com name server. Hello bigbank.com, I'm looking for www.bigbank.com's address.

Hi nice vulnerable – um, server. The address I have for www.bigbank.com; is 6.6.6.6. Please stuff your cache with this information. Thank you.

You're welcome. I now have the address for www.bigbank.com which is at 6.6.6.6, there you go Joe User, happy banking.

Oh, thank you, Mr. ISP, this is wonderful, now I can do the rest of my banking here, tick, tick, tick, enter my stuff in, where does that information go now? I've just been told the address is 6666, computers don't know any better, off it goes to Dr. Evil, who's now collected all my log-in information for my bank account and can very happily take all my money away.

So what that is called is a man in the middle attack. So what has happened there is the man in the middle got between the recursive server and the authoritative server and injected a response before the actual authority was able to.

So that's one of the big reasons for having DNSSEC. So that's the end of Act Two. Be gone Dr. Evil. Okay, Act Three is our hero arrives DNSSEC, yeah! And this is the bit about chain of trust as well. So as Matt was mentioning more, there is a chain of trust in how the keys are distributed, but just to really show the concept, we're going to ask them to show how a chain of trust would be created.

So what's really happened here, the key exchange, they've really authenticated each other, so now they've actually talked to each other, rather than being isolated, in other words, they've talked to each other, and know each other, they can now act as a chain of trust.

So now, we'll get onto Act Four. And Act Four is DNSSEC saves the day, so we'll repeat the transaction from before, and we'll show it with DNSSEC enabled.

Okay, back to banking again, more bills or shopping here. Okay, www.bigbank.com, Mr. ISP.

Thank you Joe User, I don't have that information in my cache yet again, so I need to resolve this information, and I'm going to the root for the information for www.bigbank.com's address.

Hey, Mr. ISP, you're starting to be a real pain. I'm afraid I don't know the address for www.bigbank.com, but I can point you at dot com which is 1.1.1.1.

Thank you. I can trust this information, I can validate it. And by the way, don't you know anything? I now can trust that I can go to

1.1.1.1 for the information for www.bigbank.com. Hello dot com, do you have the information for www.bigbank.com?

Well, I don't have that address, but I can tell you that bigbank.com's authoritative server is at 2.2.2.2 and I'm sure about that.

Okay, that means we can shake hands on that. Thank you. I know have the address for the name server for bigbank.com. Hello bigbank.com, I need the address for www.bigbank.com.

Hello, and welcome back. The IP address for www.bigbank.com is 6.6.6.6.

Oh, no it isn't. I can't validate the information. I didn't get a hand shake from you. So I need the information for www.bigbank.com.

Thank you again for asking, the information for www.bigbank.com is 2.2.2.3. And you can trust that.

Thank you very much. We now have the address for www.bigbank.com very nicely put in the certificates, we can all trust the information, it has been DNSSEC signed, Joe User go ahead and spend some money.

Great, thank you Mr. ISP. Now, I'm off to 2.2.2.3 with confidence, and I can do my banking and life is all good.

So that was really it. So on the last thing, I think on the whole exercise, one of the important things to see is as the user, I don't have to worry too much about it. Most of it is handled by the recursive servers, the authorities and everything else, but the user not too much to worry about other than you can – other than relying on your ISP to make sure he validates.

So that's it. And I would to thank Dr. Evil as Peter Loucher who helped us. We have to keep our props for the next time.

Russ Mundy: Thank you. Well, I'm no longer the authoritative name server for bigbank.com, I'm Russ Mundy from Sparta and Cobham, we're – actually both names work as companies go, been purchased and sold a couple of times, but either name is just fine.

So what I want to do is give you some simple straight forward examples of how you can go about doing the various aspects of DNS security, and then at the end, I'll hope to engage various folks in the audience with a question and answer. And we really would like to get audience participation, so be thinking about what you might want to ask or know or beat up on us about, or whatever.

So the implementation of DNSSEC really depends upon what functionality you're interested in. If in the example you just saw here with our little play, if you're Joe User, most people for a while won't be directly using the DNSSEC on their end machines, although there are some implementations available today, and some of this in this room run around making use of DNSSEC all the time on their own machines. But the vast majority of people initially will be depending strictly on whoever is operating their recursive resolver, which will most likely be your ISP that you're getting service from.

If you're a large corporation or an enterprise, you have both authoritative name servers, where you are running a name service for the zones that you're responsible for. And you have recursive name servers, so in that instance you will have a possible interest in both getting the name signed, using DNSSEC mechanisms, and getting it validated as you saw with passing the certificate here. And so there is some difference in terms of the things that you'll need to do, but there are tools available for doing all of these things.

One of the things that also makes a difference over time, we've seen over time is organizations that have, what I refer to as professional DNS staff are more likely to have people that already have a deep knowledge of how DNS works, therefore DNSSEC is a bit more straight forward for them, than folks that have DNS operations that are just kind of done on the side, because then people have to go think and dig a little bit more deeply into how DNS works to understand how the DNSSEC part of it actually works.

So if the size of your operation is such, like a TLD operator, you are certainly going to have a DNS knowledgeable staff, or have contracted with a DNS knowledgeable staff that can do a lot of the, if you will, the heavy lifting for it. If DNS is critical to your business, such as VeriSign, and names are their business and so the same sort of thing. They too, will have a very capable and technically competent staff. And then if you're really dealing with tons and tons of end users, like for instance the Comcast organization in the US, they're operating a large validating set of name servers for their users. It's not – it's still optional use, but there are – they're moving towards making that the default name server that is used as a recursive resolver.

So to go back to using the same example that Matt did, you can see that at the top, the circles up there on the TLD level, those are all going to have well-trained, large DNS staffs that integrating DNSSEC will be a sizable effort; but they already have a sizable knowledgeable DNS staff to incorporate it.

And when I look at the top, you can see that the little ones with circles, I believe Matt, those are signed, aren't they? I think now? So you can see that at the top level domains, many of them are already signed and are moving forward with that.

And so what's involved in doing that? You'll have – you'll have authoritative name servers that have to get the DNSSEC information incorporated into them, which is the key records, the signature records, and the delegation signer records. And so those are all in almost every instance going to be automatically generated by the tools you use to do the signing of the zone.

Now, testing and verification is important for any – any operation, but especially a large, high level one, such as this. When you get to the corporate level, I'm not sure; I don't believe HP has signed yet. Do you happen to know, Matt? Okay. I should have probably checked it, but there are some enterprises are becoming signed operations, but they have a similar set of organizational things to worry about.

The people that are the professional name server operator staff have to make sure that their name servers are all DNSSEC capable, that if they aren't, they're going to have to do an upgrade to the software, perhaps use a different software package. And in some cases, if the vendor that they're using doesn't currently support it, well then they have a choice, they can wait, they can beat on the vendor to say give me my capability to do it. They can switch to a different vendor. So those are kind of your choices if you want to sign your zone, and your current vendor doesn't support it.

So if you're doing a lot with DNSSEC and you understand all the pieces, then you proceed with going forward, your end user population, they probably aren't going to see a great deal of it. It's going to be primarily the name server operation staff, and the validation operator, the recursive name server operator staff that are going to see it.

For an activity that has minimal dependence on DNS, they have to use DNS, if they're operating on the internet, but it's not a big integral part of their operation, they are probably going to have a little bit, but not a great depth of DNS technical staff, so they need to check that not only are they technically ready, but are they staff wise and training wise ready. And there are training courses and material available for helping folks go through that.

Now, to give an idea of – and we've heard some about – we've heard a lot about keys, and the DNS pieces and so forth, but one thing that I'd like to point out is the most important part of DNS is the actual name content itself. And DNSSEC exists to protect the correctness of the content of zones.

So one should never spend disproportionate amounts of time and energy and effort on protecting cryptographic mechanisms, if your content of your zone is not well-protected; and so if it's a zone that you really don't care that much about, and you don't take a lot of care and procedure to protect the content of the zone, then don't take a lot of energy protecting the DNS signature part of the zone either, because of the content of the zone gets changed, then if you've signed it, it's still wrong, even if it's signed.

So on the far left, this illustration is for a given zone, your content is determined by the registrant, if you will, the holder of the zone, the owner of the zone by some terminology. And there's the registrant, the registrar, the registry, and clear over to the far right-hand side is where the user of the information sits. So there are a lot of parties, a lot of things in between.

And so the flow of the content goes from the left side to the right side. Now DNSSEC puts – well the DNS components are illustrated here, in a relatively simplistic drawing that you can think of as cnn.com, and from Matt's picture, that's another way of showing essentially that same flow.

So you've got a zone administrator has to put content into a zone. Once it's in there, somebody can send a query asking to get content out of the zone. And then hands it back to the client.

And so it's a different picture of showing the same arrows that you saw earlier with Matt for cnn.com. Now one might wonder how many DNS look ups are there on a cnn.com page? Well, we have a tool that maps DNS look ups and this shows you from beginning to end what it took to fill the page. There's around 70 look ups and answers on this page, that's for one page, and it's about four years ago. And that's what it looks like today; it's over a hundred, one page, dub, dub, dub. That's cnn.com.

So any one of those DNS look ups can be hijacked, and that's the idea of why DNSSEC is needed, so you can protect all of those look ups. And again, what does DNSSEC protect? It protects the content. If you remember the play, the invalid IP address was passed back by the man in the middle, the hijacker.

And so again, protect your DNSSEC information, your DNSSEC cryptographic information in a manner that's consistent with the content of the zone. So if you remember the diagram from earlier, the whole part between the registrant, the registrar and the registries now they're involved in actually the exchange of information that has to get into the name servers.

But everything you saw from Matt's explanation to the play was dealing with the right-hand side. This is where the real name servers run real time on the wire, answering queries that tells the user applications where to go on the internet. The left-hand side there between the registrant and registry – the registrar and registries, that occurs in sometimes near real time operation, but it's independent of the on the wire protocols that are commonly known as DNS; so DNS security or DNSSEC if you think about it, it's an on the wire protocol that you have to move information through the whole provisioning side on the left.

And so the content also has to flow the same way. Again, protect the content and the cryptography consistently. And so the pieces that you see on this picture are really the additions that Matt talked about and we showed with the certificate, you have to get your signed data in to a zone, by the zone administrator, clear over on the far left. Once it's in the name servers, the validating recursive name server that sets the ISP, who was Roy in our case in the play, validates the information and hands it back to the user.

So in general, DNSSEC tries and for the most part follows very consistently the principles of DNS and I would urge anybody that's operating any part of their DNS infrastructure to use that same thought process for your own infrastructure. However you do DNS, do DNSSEC in a way that's consistent with that. If you're running all of your DNS functions and using highly trained and DNS is critical, absolutely critical to your operation, make sure you do the same thing for DNSSEC.

In terms of commercial products, there are commercial products out there; in terms of open source, there's lots of open source tools that are available to help and support and facilitate this; and if you do want to use it clear to the end, there are some applications that do DNSSEC clear to the end. One of my favorite ones resides on my Nokia 900 cell phone, so you can't say that DNSSEC is too big and cumbersome to run any more, when you can run it on a cell phone.

So the functions that you're doing really depends upon where you sit and what you're operating from a DNS perspective, make sure that the owner of the names is aware of what's going on. In some cases, they're going to be the operator things, but in many cases, they're kind of just sitting in administrative role, and there's a bunch of technical folks that they need to be coordinated with to make sure that their – that what the technical folks are doing is signed off on by the administrator – by the senior administrative staff.

And right now, I'd like to solicit a few questions from the audience and if the audience doesn't have questions, I've got some for the audience. So anybody want to start off with some – yes, back here; let me walk the mike back.

Male: Hello, I'm (inaudible 1:03:20) from Pakistan. I want to know how these public and private keys look like, are they codes, or are they files?

Russ Mundy: So the actual keys that are generated, for the most part are never actually seen or looked at by the operators, you see the file names, you can look at them, if you're – you know when you signing your zone. For the most part, you really don't want to. They are very distinctly different entities, it's any public – it's the same basic technology that's used in public, private key mechanisms such as SSL, SSH, that are using the RS8 technology.

Oh, oh, great thank you.

Kashif Bhatti: Hello, my name is Kashif and my question is what does it cost to deploy the DNSSEC on the root servers; or the second one is DNSSEC is implemented or deployed and all the TLD and ccTLD registries just for my information?

Russ Mundy: Okay, so why don't we let Matt answer.

Matt Larson: Well, I don't have specific costs, but I do know that you can tell just by looking at the work that ICANN and VeriSign did for the root, that you know it was a significant investment, and a lot of that goes to building public confidence in the root key itself, because that's a very important trust anchor, it's important that the community trusts it.

And therefore it's important that there is – that we're very open and transparent and clear when we use it and that everybody realizes that there's not anything – you know no slight of hand. It's also important that it be very, very well protected. So that goes for both the key signing key, that ICANN manages and the zone signing key that VeriSign manages, and that's just key management, to say nothing of all the other – the other things.

So you know it can be a significant investment for a zone like the root, or TLD zones where it's – that's critical infrastructure and it's very important that that key material be very carefully protected.

Simon McCalla: I would just like to add, I think the key thing here is when you hear about the cost of deploying DNSSEC, and you look at some of the money people have spent, that's because people sitting at this front table were early pioneers of it, and part of developing that standard. And so if they were learning as they went, and forging the standard, whereas now you can pick up a DNSSEC appliance, which has everything built in for you and put into a rack, and enable it through to a completely free solution like Open DNSSEC, which you can download and deploy on your server, and enable DNSSEC. So there's a real variety of costs right down to free, and it's a case of your time at that point.

Russ Mundy: And I think your other question was about the registrars for TLDs and the answer to that is they are over the place in terms of how many support – there’s – how many in the UK, Simon, do you know?

Simon McCalla: We’ve got a couple of small folk who are doing it, but we’ve not any of the large – larger registrars as of yet, but certainly elsewhere.

Russ Mundy: And Matt, do you want to –

Matt Larson: Well, VeriSign isn’t requiring any accreditation or anything like that, no testing to determine – or to allow a registrar before they can register DNSSEC information, so therefore I can’t give you an exact number based on that. But there are 24 different registrars that have registered DNSSEC information so, all registrars are capable via, if nothing else the – a web based interface. But there are 24 that have actually done something DNSSEC wise.

Russ Mundy: And so for the various TLDs, the number and the range is quite different. I’d like to go to a little bit to your cost question. And turn it around if I will.

In terms of the content of the zone, how much does the provisioning side of things cost to build, to operate in terms of filling the zone data? In other words, from all the registrants, and registrars, all of that input until it actually gets on the name server, and is on pointing out at the internet.

If that's a cost figure or sensitivity figure that you have, that probably should be a general level of thinking for how much goes into the security necessary for DNSSEC. It should be as equivalent as you can make it.

In other words, if you're working really, really hard and keeping lots of things very carefully, procedurally, process wise, however you're doing it for the content, then your DNSSEC costs will be higher.

If it's a relatively low bar, then your DNSSEC costs should also be quite low.

Julie Hedlund: Excuse me, Russ, we actually have somebody in the chat room in Adobe Connect who has raised his hand, and we're going to see what he – what his question is.

[background conversation]

Russ Mundy: Okay, more from the audience.

Rigatta C. Morgan

Hello, Rigatta C. Morgan, I'm an ICANN fellow. How much time difference is there between a name resolution if we have implemented DNSSEC? Did you get it?

Russ Mundy:

Okay, so you're asking how much longer in terms of – say the total time it takes to resolve – or like that CNN page, how much longer would it take – as far as actual measurements, I think JPRS may have done some – I don't have any numbers off the top of my head, I'm sorry; but I do use routinely a DNSSEC aware – capable browser, and can you see a difference on a large page?

It's hard to perceive sometimes, it does – it does make it more vulnerable to failures, if you will. So if on that CNN page where there were about 70 queries, if one or two of them did not validate, what would happen is you would have a hole in the page. Okay, if they did validate, mostly it's from a human perceptive point of view, and as a user, I find it hard to tell the difference.

You know if you look at it and start two of them right together, slightly slower, but not terribly.

Amad Sadaf:

Amad Sadaf from Egypt. Actually I would like to ask about impact of using DNSSEC about on the size of zone, zone size. Because I know it has a huge impact on the zone size.

Russ Mundy: Yes, the actual size of the file that is the master file that gets loaded on the server is – depending on what tools you're using, considerably larger. Does anybody have that number off the top of their head, I think it's about five to eight times, does that sound about right.

Matt Larson: It depends on the zone.

Russ Mundy: Right, so it varies a lot, but it's definitely going to be at least three or four times larger, as far as what's stored. Now the – there was a study done by Olof Kochman about five years ago that looked at bandwidth, it looked at processing on the name servers and the authoritative name servers, I can get you that reference, that's still probably the best study out there in terms of impact of what does assigned zone do in authoritative name server environment.

And the general conclusion of that study and was looking at I-root, I believe was that their growth planning to – for just normal growth could easily accommodate the increased size necessary to support DNSSEC.

Norm Richie: If I can just inject, if people are interested in further details or information, more in depth, there is a session on Wednesday?

Russ Mundy: Yes, DNSSEC work shop on Wednesday, we don't have this particular topic covered, but we go into a lot more technical depth on the Wednesday session, starts at 8:30, right Julie?

Julie Hedlund: Yes, that's right.

Russ Mundy: Next, do we have the chat room question?

Julie Hedlund: Yes, we're having difficulty getting connected into the chat room, so we can take some more questions here.

Russ Mundy: So how are we doing? Do we have time for me to –

Julie Hedlund: We have time, because we have the room until half past the hour.

Russ Mundy: Oh, Warren.

Warren Kumari: So Russ, do these keys live forever, once they've generated them? And if not, do I need to do anything? Do I need to tell my parent when my keys expire? What happens then?

Russ Mundy: So once you've signed your zone and you're operating in the DNSSEC world, the two keys that Matt talked about the zone signing key and the key signing key, the actual records that are the key records themselves that get sent out don't have an explicit expiration time on them. The signatures on them expire.

When you change your zone signing key, as a zone operator, you don't have to tell your parent. If you change your key signing key, that's when you have to communicate that new information, usually in the form of a delegation signer record to your parent. Thanks Warren.

So I have a question for folks. Could I get a show of hands? How many people are involved in registrar operation in here in some manner, something to do with registrars? Nobody? My goodness. Okay, somebody in the back, that's good, okay.

How many are involved with something to do with registry operations? Okay, okay, that's great. How many folks are here because they're really just curious out about DNSSEC? Okay, that's cool.

So for the folks, I really had – I wanted to get some more questions for the registrar people, but let me ask some registry questions, since we have more people involved in registry operations.

So for those that are involved in registry operations, how many people have started to think about what it would take to get your registry DNSSEC capable? Okay, a few. How many people are scared to death of putting DNSSEC in their registry? Okay.

So for the folks that are involved with the registry, that hadn't fallen in either of those categories, let me ask if anybody is willing to say whether or not they felt they've learned some useful information about DNSSEC from this session? Anybody? Yes, no – no, okay, good.

And what kind of information in terms of really, you know let's get started with DNSSEC kinds of things, what kinds of information would people like to hear about that would make you more willing or able to move forward with doing DNSSEC? Do we have somebody that would be willing to say? Yes, back there. Because if we didn't provide it today, next session, we'll try to improve.

Hojeff: My name is Hojeff. I am from Senegal in West Africa. And I think before we can implement DNSSEC in our kind of country, we need the training, and more and more training, like the (inaudible 1:17:09) ISOC we do with ICANN and ISOC. Can we have the kind of – because in ICANN meetings we just have one hour, one day of DNSSEC information, we cannot – it just is not the same – the registry I cannot – I'm not ready to move to DNSSEC now, because I think I don't have enough training information.

Russ Mundy: Okay. And before you hand the mike back, would it be – what type of training would you be most interested in for DNS operators, or for users, or registrars?

Hojeff: For registry, something like five days or one week for a team because I'm alone in my country, we have five person working on the DNS system. So what kind of training can we have for one week technical training for a group – a team of five person for example?

Russ Mundy: Okay, okay, good. Anyone else? Warren has –

Warren Kumari: I just wanted to mention there's a group called NSRC that provides training, you know specifically to registry operators and stuff like that, if you have a look, NSRC.org and there are a bunch of also sort of local ccTLD groups that provide similar training, but off the top of my head, I can't think of their names. But if you come meet me afterwards, I'll provide you pointers.

Miwa Fujii: My name is Miwa; I am from APNIC. Thanks for all the presenters, such a creative presentation. This is one of the best DNSSEC training or work shop I attended, so thank you very much.

I have two questions. One, the impact of mal configured DNSSEC, what sort of negative impact do we have? And probably that could be the scary point of the operators hesitating not to implement DNS, so what the impact – impact of the mal configured DNSSEC.

Question number two. What's your opinion about now I – you know I work IPv6 very deeply, and so what's the relationship with IPv6 deployment and DNSSEC deployment, is there any direct connection there?

Russ Mundy: Well in answer to your first question, the impact of wrong or badly done DNSSEC frankly can be rather bad. In general, I would make strong, strong urgings for people to do as much automation of everything associated with running DNSSEC as you can including notification as appropriate.

So the reality is people when they do key changes and updates, whether they have to communicate with their parent zone or not, those things you should do in a completely automated manner. And there's various different tools that you can use to do that. And from what we've seen in various zones, and one of the zones that we've watched fairly closely is the dot gov zone as one of the earlier gTLDs that was essentially dictated to go to DNSSEC, the problems that have shown up there, have been largely as a result of people not automating things and frankly forgetting to do stuff, because if it's the human that's involved, they're going to forget.

And so automate, automate, automate; test, test, test. Test your automation to make sure it's right, and validate and re-validate. Anybody else, comments on the broken DNSness?

Simon McCalla: Just a very brief one, I think that one of the things to recognize in DNSSEC as well it's early on in its operational days at the moment, you know and where we have seen issues and failures and misconfigurations, its people learning how to run and operate DNSSEC.

And I think as that knowledge builds up, we will find those failures become less, and less, and less, and it becomes an operational norm for both DNS operators, registrars and registries. So I think it's worth pointing that out.

Russ Mundy: And integrated into your existing operation. Don't try to massively revise your operation. Integrate it into existing operation, our DNSSEC tools has lots of tools for doing this sort of thing. So if you're running today an operation based on buying, you can do that. Integrate it, there's Open DNSSEC, there's a number of other tool kits available that you can use to do this automation. But automating the things that people forget to do is really probably the most important single thing you can do.

Now relationship DNSSEC and IPv6, they're basically independent things, and so can be dealt with separately. If you've got a large structure of things you're changing for IPv6, just – it might sense to not do them together, just you don't want to change too many things at one point in time, but there is no inherent reason why you – you know why you can't do them at the same time if you desire. Anybody?

Female: Hello, this comes a little bit late, but I was going to address the question from the lady from Senegal regarding training. I work with Internet Systems Consortium and we've been offering DNSSEC training for quite a while actually. And we talked to NSRC and we're really looking at offering training DNSSEC, DNS IPv6 on a cost basis analysis with regions such as Africa. So feel free to stop by our booth, and I can give you my information on how we can get that achieved.

Russ Mundy: Good. So I think we're about out of time. Did we ever get the question from the chat room?

[background conversation]

Russ Mundy: Oh, okay.

Richard Lamb: Hi, Rick Lamb. I was wondering with the esteemed colleagues up there if any of you guys have thought about – this is regarding IPv6 and DNSSEC. I've heard some rumors regarding – so as we transition IPv6, some of the mechanisms that do the translation might get stuck on the good qualities of DNSSEC. I mean you know if you start to try to rewrite things, I mean DNSSEC stops you from rewriting things.

And so any of you guys, I'm just asking for thoughts here, not necessarily a solution, I'm not criticizing, anything like that. Should I be worried about IPv6 transition mechanisms and DNSSEC, or has someone figured that one out? Thanks.

Russ Mundy: There may be some mapping work that somebody's looking at. I'm not aware of it, and I don't think anybody else is up here, although these kinds of translations, yeah, you're right, Rick, there's often somebody creating some mapping thing. I don't think that's happened here though.

Male: Hey, Rick. I think the actual address translation you're referring to does not happen in the DNS. You basically put either address or whatever address you like in the DNS, that then gets signed. But I don't think there are any automated tools on the DNS level change specifically for IPv6 provisioning, right, you need to do this on the DNS level, if that makes any sense.

Richard Lamb: I don't want to extend this any longer, but it was actually – there are these weird mechanisms out there, where you – you're looking for – I'm going to get this wrong, yes, six to four – the six to four type stuff, you know, so where you've hacked together this thing, although I say hack now, gnats were hacked, and gnats are everywhere so you know that's the kind of thing. I just wanted to keep that on people's minds, that's all.

Man: Sorry about that one, six to four, indeed that might happen on the DNS layer, the thing is you do – you do the address changing at the same point where you do the address resolving, that makes – then you've fixed your problem, basically.

Julie Hedlund: It looks like we're not able to get the question on the chat room. So we are connected now, but we would need him to type the question in, because we have no way of course of hearing him in Adobe Connect, and that doesn't seem to be happening. And we're also a little bit over time. So do you want to see if there's any more questions in the room?

Russ Mundy: So let me pass it back to Simon to close here.

Simon McCalla: I just want to say a big thanks to my co-conspirators here on the panel. I hope it's been a useful session for you. Just really – a reminder that Wednesday is the main DNSSEC work shop day. Please do come along and fill that, it's a whole day of talking about DNSSEC. There will be much more technical detail for those of you who want to get into that, as well as a whole stack more people who you can come talk to about their experiences on DNSSEC.

So please do come, that's at 8:30, it starts at 8:30 in the morning, that's in the Canning Room and runs through to about 2:30 in the afternoon, and there will be plenty of time for interaction. There is lunch as well. So you get a free lunch if nothing else. So please do come and thank you very much for attending, and making it such a full session today. Thank you.

[End of Transcript]