KOBE – DNSSEC Workshop (2 of 3)
Wednesday, March 13, 2019 – 10:30 to 12:00 JST
ICANN64 | Kobe, Japan

[JACQUES LATOUR]: Okay. Good morning. So we're in the second part of our DNSSEC Workshop, and we have four presentations. The first two are on DNSSEC and DNS-over -TLS/DNS-over-HTTP. So that'll be the first topic of discussion. Then the second topic is about KSK rollover information.

So next up is Warren Kumari from Google. Warren?

WARREN KUMARI: Hi, all. I'm Warren. So apologies in advance. This might be slightly rude, seeing as this is the DNSSEC Workshop, but what I was thinking is this DNSSEC thing is kind of hard and annoying to set up, and it's a bunch of work. And it's also, like, 20 years old. There's a whole bunch of new protocols, like DNS-over-TLS and DNS-over-HTTPs. If I've got those, do I really need to bother doing all this DNSSEC stuff? Do I really need to bother with all the additional work and faff and all of that?

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.

So let's talk about that. First off, a disclaimer. This is an introduction. There are some simplifications and, in some cases, oversimplifications. We can shout about those later.

Let me quickly start my timer.

So the first and most important thing to know here is that confidentiality and integrity protection are two very different things. So confidentiality –

UNIDENTIFIED MALE: Warren, [get] closer to the mic.

WARREN KUMARI: Closer to the mic? I'm not tall enough to reach – there we go. Is this better?

Okay. So confidentiality and integrity protection are two completely different things. Confidentiality is keeping a piece of information secret – and now the sound is going weird. So that's sort of encrypting something – let me try the other mic because that's [inaudible].

Is this working? Nope – yay! Okay. And integrity protection is being able to verify that a piece of information is correct. So this is sometimes a bit of a tricky concept to understand, so here's an example. This is an ATM receipt. The number which is circled over here is the balance. This information I probably don't really want to share with the whole word. Hello, whole world. But it doesn't actually really, really, really require confidentiality. It's no the end of the world if people know what the bank balance is.

But what I do really want to be able to make sure of is that I can verify this bank balance. If know what my starting balance was and I've deposited some money and then I've withdrawn some money, I need to be able to verify that this information is correct, that the bank and I did this calculation and we both end up with the same number.

So that's an instance of where I want something like integrity protection or the ability to do verification.

Confidentiality is more something like the PIN number for my ATM card. It doesn't really matter that much how I came up with the number, but I want to make sure that it's confidential and that me and only me know.

So a little more of this. Another example of integrity versus confidentiality. For an election, I really care about the integrity of the outcome. I don't care about the confidentiality. In fact, confidentiality for an election would be fairly useless. By definition, you want everybody to know the outcome of the election. But the important bit is that you can verify the integrity, you can verify that the person who got elected was the one who should have.

Where the confidentiality comes in is my particular vote. I want to keep that confidential. I don't want anybody to know who I voted for. For the security geeks in the room, this is an obvious other important thing: object versus channel security. A very quick overview. Object security is something like taking a piece of information and encrypting that or providing security of that. So an example would be if I were writing a letter, I could write it in some sort of secret code and that would be object security for that information.

Channel security is more taking a piece of information or taking a letter and putting it in a tamper-proof envelope and sending it through the mail. Channel security keeps something secure through a specific medium.

So how does this relate to DNSSEC and DNS-over-TLS? So what DNSSEC gives you is it gives you integrity protection. So this thing up here is public information. That's one of the signatures for the IETF.org domain. It's a piece of public information, and anybody can have a look at it. Anybody can verify it. There's no confidentiality in that.

However, using this public information, you can do some cool things. You can make sure that the information has not been modified on the primary DNS servers or the secondaries after it's been signed. You can make sure that it hasn't been modified at all on the path from the Internet to the ISP. You can make sure that the ISP has not been tampering with it. This is of course all assuming that you're doing validation on your laptop. You can make sure that nobody has modified it. You can make sure it hasn't been modified on the link from the last mile from my house to the ISP. Also, I can make sure that it's not being changed in my network. There've been a number of cases recently where home routers, home [CP], have been compromised, and people have been making changes to records there.

However, DNSSEC does not give you any sort of confidentiality. What this means is an attacker who is on the Internet somewhere between my resolver and the Internet can see exactly what it is I'm looking up. That's not quite true. They can tell that somebody at the ISP is looking up a set of names.

My ISP, on the other hand – wow, that got louder – can actually tell who is looking at the name. So they can see that I, for example, am looking up a name. They can tell what I'm looking up. This is also true for an attacker who is somewhere between my house and my ISP. They can see what I'm looking up. Also, once again, within my home network on my home router, etc., an attacker can look up and can see what I've been looking up.

This doesn't necessarily sound like that big an issue. "Eh, it's DNS traffic. Who really cares?" But what a user looks up actually leaks a huge amount of information about that. So, for example, if I'm looking up AlcoholicsAnonymous.org, chances are I'm looking it up because I'm trying to find an Alcoholics Anonymous meeting. And I'm no longer really anonymous.

Another good example is if I'm looking up GayRights.org. The fact that I'm looking up a name has some implications for my privacy, leaks some information about me, which I possibly don't want to do.

More worrying things is, in some countries, if you're looking up different political parties, this is potentially something that does not end up well for you. Also, obviously, anywhere where an attacker can see what you're looking up, the attacker can also block what you're looking up. They can implement censorship and they can stop your ability to look up names.

So, this is where things like DNS-over-TLS, DNS-over-HTTP, DNS-over-DTLS, and [D-Dash-over-Fu] come in. What these things provide is confidentiality for your lookups. So what they do is they provide a means so that, on my home router, on the link from my home router to my ISP, I get confidentiality of my lookups. Nobody can tell what it is I'm looking up.

Unfortunately, on the resolver, that information is still visible, so my ISP can see what I'm looking up.

Then currently also from the resolver going out to the public Internet, this information is visible. Attackers can see that somebody is looking it up if they're outside the ISP; within the ISP, who is looking something up.

The IETF is currently working on improving this so that these links will also be encrypted. They will be DNS-over-TLS or something similar from the resolver to the authoritative name servers.

Actually, let me quickly go back. So in a number of cases, censorship is happening within a country. This is a relatively famous picture. Back during the Arab Spring, Turkey blocked access within their country using their local country resolvers to a lot of sites that people in Turkey potentially wanted to see. This was spray-painted on a bunch of buildings, saying, if you'd like to get around the censorship, if you'd like to be able to see what's going on, change your DNS to instead use this outside the country. That worked for a while. Unfortunately, it got blocked relatively quickly after that.

This is a picture of how that works. If you have something like DNS-over-TLS, you get encryption from your house, so nobody who's owned your CP can see it, nor on the last mile, within your ISP. Hopefully, you're talking to public resolver outside the country if you're worried about censorship. This is entirely your choice. You should be able to choose if you want to use your ISP's resolvers or whatever resolver you would like. Unfortunately, still, on a whatever resolver you're using, your query visible. But then on the last mile, hopefully it will be encrypted again.

So this is DNS-over-TLS. You get some confidentiality for your lookups. But what you don't get with DNS-over-TLS is you don't get any protection against the records being modified on the DNS servers before they get sent to you. A lot of people outsource their DNS to a provider. If your secondary name servers are not run by you, you potentially don't trust that nobody has modified them.

You also do not get any protection against somebody modifying the records on the resolver itself. So whatever resolver you're using could rewrite your answer if you're not using DNSSEC.

So the obvious questions is, why don't we have both? This is a picture showing what you get if you use both DNSSEC and DNS-over-one-of-these-new-protocols. What you get is you get confidentiality fairly much all the way along the path. You also get the added advantage of integrity protection and strong integrity protection so that you yourself can verify that the answer that you have gotten is actually the one that you were intended to get by the person who runs the zone.

Basically, another summary of this. DNSSEC gives you one set of protections. DNS-over-TLS gives you a different set of protections. If you actually want real security, you should really be doing both.

So, unfortunately, my initial rant of "This sounds like a lot of work. I don't want to have to do both" ended up not being right. I have to do both if I want good security.

Questions? Hopefully there are some.

[JACQUES LATOUR]: Thank you, Warren. Any questions?

WARREN KUMARI: None?

UNIDENTIFIED MALE: Oh, Russ has one.

RUSS MUNDY: Thank you for the presentation. I do have a question for you, but before that, I want to remind folks that, especially for interpreters and the recording and the transcript, say your name first. This is Russ Mundy.

Okay. Thank you, again, Warren. When one does this combination of technologies, what do you see as the possible pitfalls other than it's harder to do both? Are there holes you can fall into when trying to do both of these?

WARREN KUMARI: I think the primary one is it's more work. You have the risk – any security thing that you add adds some addition risk of it failing. So what you have done is you have added up both sets of risks, so there's more potential for things to go wrong.

However, the protections are there for a reason, and I think that that tradeoff is well worth the risk.

So, yeah. I don't know if that actually answered that.

JACQUES LATOUR: So I have a question. So DNS-over-HTTPS today. Does it have the mechanism to check the integrity of DNSSEC?

WARREN KUMARI: Nope. They're basically orthogonal. So DNS-over-TLS/DNS-over-HTTP is largely just a transport for DNS. Think of it as a VPN for your DNS along each hop.

So the DNS, once it's wrapped in TLS, cannot be modified in flight without TLS having some issue. But it can be modified at the start point, at the resolver, or at the end point.

Did that answer your question? You like as if I answered a completely different question.

JACQUES LATOUR: So does the implementation in, like, Firefox today – does it validate the DNSSEC?

WARREN KUMARI: As far as I know, no. It does not currently do that.

JACQUES LATOUR: So we need to work on that.

WARREN KUMARI: So you should work on that, yeah. And I would think that it's reasonable for implementation to do both DNSSEC and DNS-over-TLS. Actually, Vittorio, I think, is going to have a good presentation in a bit on DNSSEC versus HTTPs: Who Do You Trust?

And I would suspect also – correct me if I'm wrong – a thing on Do You Trust Your ISP or Someone Else's (a Third Party)?

JACQUES LATOUR: Thank you, Warren –oh, sorry. Question. One more.

JOHN: This is less a question than a comment, but it took me a while to appreciate that one of the important advantages of DNSSEC is that you no longer care where your data is coming from, and it enables a sorts of – hyperlocal root. Basically, it enables all sorts of funky DNS technology, but so long as the signature is good when it arrives, it doesn't really matter what bizarre route it took to get there.

WARREN KUMARI: Yeah. So I wasn't sure how much time I would have, so I didn't go too much into that. One of the things that DNSSEC provides is the ability to do things like DANE and similar mechanisms. For those sorts of mechanisms, you really, really, really, really want to be sure that the answer you got was the answer you should have gotten. You really do not want to be trusting the resolver to make these security decisions for you. So, yeah, that's pretty much exactly what you said. This allows the creation of other things, where the information between the zone owner and yourself is critical and you don't want anybody in the path or along the path to modify it.

JOHN: Yeah, but the point though is this enables all sorts of complicated, untrustworthy paths, just because – it's sort of like TCP. It's like, who cares what's in the middle so long as you get the right bits at the end?

UNIDENTIFIED MALE: All right. Last question.

UNIDENTIFIED MALE: I just had a follow-up on what John said and what Jacques asked. The DNS-over-HTTPS will let you – you could validate the certificate and know that you're talking the DNS server you think you're talking to. So you can get that.

But as John said, it doesn't really matter. What matters is the object security of what you get back.

UNIDENTIFIED MALE: All right.

UNIDENTIFIED FEMALE: [inaudible] remote.

UNIDENTIFIED MALE: Remote.

UNIDENTIFIED MALE: Okay.

UNIDENTIFIED FEMALE: This is from Geoff Huston. The question is whether there's a difference between using TLS as a transport for DNS transaction and the use of DNS as HTTP objects.

WARREN KUMARI: Yes, but I think answering that would take too long here. I think that's more of a fight Geoff and I can have later, or a discussion we can have later. Unless – does anybody have a good answer for us [inaudible].

UNIDENTFIED MALE: Well, let's do Vittorio first.

WARREN KUMARI: Okay.

JACQUES LATOUR: Then we can … All right. Thank you, Warren.

So next up is Vittorio Bertola from Open-Xchange.

VITTORIO BERTOLA: Thank you. Well, while you bring up my presentation on the screen – okay. Thank you. I'll say that, more than a presentation, this is a set of tools that came up in the last few weeks while being deeply involved in the discussion on DNS-over-HTTPS because – I didn't talk with Warren, so I didn't know what he was presenting, but he did a good job in presenting the concepts. Everyone who's more or less knowledgeable about DNS ends up with the conclusion that you need both. These are two separate things.

But then, when you discuss this people that maybe are less involved with DNSSEC, they seem to think that, in the end, maybe you don't. So I started thinking, "Is there some merit to this? What's exactly the difference? Are we actually sure of this conclusion?" So I will go very quickly on the introduction because Warren already did this.

The points I wanted to make related to the difference between DNSSEC and DNS-over-HTTPS is they are actually very different things but also they are very different in the way they were conceived, and I'd say in the moment in which they were conceived.

So DNSSEC – I was not involved in designing DNSSEC, of course, but apparently one of the key requirements was to find a way to ensure that integrity without having to encrypt all the communication because, at the time, I guess that was considered too much of a problem, especially in terms of computational load.

So, seeing with the ISO today, maybe you wonder, “Why didn’t they just encrypt the communication?” like the DNS-over-HTTPS or DNS-over-TSL they’re trying to do today because that’s the model that basically has been affirming itself in the last ten, 15 years. But possibly at the time, there was a stress on not actually doing that for – this is what I gathered from the comments of some people.

So in the end, if you go through what this does, it tells you that you can trust that the reply was no altered. So the objective is that you, as a client, receives something, some data, and you can trust that they were what you actually wanted to get. They have not been altered by other parties in transit, whether it’s the resolver or it’s someone on the path. So in the end, this gives you data security or object security as well, I would say.

DNS-over-HTTPS is different because it basically encrypts the communication, so it encapsulates the DNS-over-HTTPS communication. So the entire communication is encrypted, so this provides confidentiality different from DNSSEC. Of course, it requires encrypting everything. It has some computational cost, but today this is not a problem anymore.

So in the end this gives you channel security. So it basically secures the channel you used to communicate with your resolver. So in the end, people say, “Okay. So it seems the channel is secure. No one can mess up with my reply. So this is also providing me a guarantee that the reply was not altered.” So in the end, this is the same thing.

So people actually start to say, “In the end, this is a different security mechanism, but in the end the final result is that no one can mess up with the replies to my queries when they are being sent back to me.”

So why can’t we just use these encrypted protocols in place of DNSSEC? This is actually a very appealing proposition because, if you say that the level of security is the same, then DNS-over-HTTPS has several other advantages because then it makes your queries private. It gives you confidentiality. It [can] authenticate the server you talk to, which is another problem that has not been solved with the previous technologies. In the end, it’s easy to implement, especially for applications. So people actually say that, “No one could make DNSSEC really work and get it adopted in 15/20 years, but HTTPS is everywhere, so we can just go for that.

So of course this is not really true. So there is fallacy in this idea. The point that I wanted to stress is that the difference lies in what can you trust and what's the source of the information that you can trust? In DNSSEC, it gives you security on the fact that the reply was not altered throughout all the change, so starting from the authoritative zone in authoritative server that serves it throughout the entire chain up to you, while in the end, DNS-over-HTTPS only gives you security with respect to the resolver. So the only thing you get is that, yes, the reply was not altered [irrespective] from what you got from the resolver.

So of course the conclusion you can get to is that, if you want to get better data security and channel security [inaudible] confidentiality and integrity, you need to have both.

But in the end, do you really? Because it's just a matter of who you trust. So the thing that is changing in these two models is that, in DNSSEC, the trust and the source of the [truth] is the root server system. So with everything you get, all the replies you get that are properly signed and validated through DNSSEC, it's basically the root server system that tells you that these are the truth. While with DNS-over-HTTPS, you're just accepting whatever you resolver tells you. Since it has not been modified, this is the truth. So as long as you accept that the truth is whatever your resolver is telling you, then you're fine with DNS-over-HTTPS. You don't really need to do DNSSEC validation.

So the problem is that, what's the truth today in DNS? Of course, DNS was conceived as this sort of distributed database in which would there be truth. So there would just one correct reply to one DNS query, and all the rest would be false.

Still today in these discussions, there's plenty of people talking about DNS lies and this kind of terminology, which is understandable. I'm not challenging it. But the reality is that, in fact, already today, the replies you get to your DNS queries are heavily dependent on who you are and where you send them, so which resolver you are using.

Okay. So in the end, the DNS community often tells you that this is just a shortcut. So, yeah, there are like, 20 different reasons for which resolvers change their replies they send you. One is censorship. It's the most heavily mentioned, but actually, [in a sense] HTTP just does more a minority of the use cases that are based on modifying your replies at the resolver level. Many of them are related with security, so there's all these split horizon and the local names and stuff that is being done by your network administrator at the resolver level to improve the security of your network to draw a parameter and ensure that the data should not go outside of your network are not exfiltrated and for all these purposes.

There are other cases which are related to, actually, voluntarily not being able to go to places that you don't want to be reachable from your network, whether it's a company that wants to prevent employees from connecting to Facebook during worktimes or whether it's family that doesn't want the children to get inappropriate adult websites.

There's even cases in which the government is blocking websites but it's not for censorship. It's for other reasons, like not paying taxes or this kind of stuff.

The there's CDNs (Content Delivery Networks) that are often based on modifying and giving you a different reply depending on who you are and where you are.

So in the end, one of the thoughts I'm starting to have is, can we really continue to say that the DNS is a single distributed database and the DNS query and reply mechanism is just a way to [read] from this database? Or is the DNS becoming something much more complex? So it's a service where actually the mechanism can foresee some kind of policy or some kind of localization of the replies and so has multiple levels of complexity. Because if you start having this question, then the question of what's the truth is really relevant.

So, if the question of what's the truth in DNS becomes relevant, also the discussion on what should be the source of the truth becomes relevant. Then you can actually discuss whether you do need to have data integrity going back throughout the entire chain and through the root server system.

So in the end, the point is that the already today you are already expected to trust the resolver because, in the end, there are, I think, very few systems that do DNSSEC verification on the device. So in the end, must just rely on a resolver doing the DNSSEC verification for them. And if this is the model, then you already have to trust that the resolver is doing it and it's not lying to you because you're not actually checking that what they get back to you is actually DNSSEC-valid.

The picture that Warren showed with integrating confidentialities is pretty good, but the requirement is that the DNSSEC validation is done on the device. And if we move to this new, let's say, model in which a replication uses DNS-over-HTTPS to connect directly to maybe a different resolver, then each individual application has to implement DNSSEC validation. It's not even a matter of implementing it in the operating system. It's a matter of having each and every application that makes DNS queries also verify whether they're true, which I think would be very good.

But is this something that is realistically possible? If so, how can we get there if this is the way?

But this maybe could also not be the way because, in the end, if we accept the model that you have to trust your resolver anyway, then it could be enough to have DoH and authenticated connection to a resolver. Then the resolver maybe could use DNSSEC for the rest to verify what they get from the authoritative servers. Maybe it would be fine.

Then of course, there's the disruptive case, which is also one of the concerns in all the DNS-over-HTTPS – sorry. I think the translators will be getting [crazy].

Then there's the disruptive case in which DNS-over-HTTPS becomes a way for the resolver operator to actually own the namespace. So if you go to the model in which the resolver is the source of truth, then the resolver becomes able to actually decide what they can tell you. It could even not use the root server system. It could just make up domains, make up TLDs.

While it has been pointed out that this is already possible and has not happened, I'm not actually ready to accept this. But this is one of the concerns with the new DNS-over-HTTPS deployment models. So this is another case that could happen.

So, as I said, this was just a set of questions [inaudible] more than anything really coherent. It was meant to spark some discussion. In the end, if you think of these, if you put yourself in the user's shoes, yes, we all trust the root server system, but if you ask the user, “Do you trust ICANN? Do you trust this [world]? Or do you trust maybe the maker of your browser, which may be a very trusted non-profit entity? [mention one], do users really want to get, let’s say, ICANN to [bless] the reply to their DNS queries, or are they [enough] with another entity that they trust the resolver in so it’s fine for them?

And the other question is, since one of the topics for this workshop was how do we get browsers to implement DNSSEC, is there really any reason for browsers to do this? Maybe they can just do DoH and make sure that the resolver implements DNSSEC.

So, again, I think that there is some discussion that needs to be done on whether the original model for the DNS is still valid today. I think that the example of the requirements for DNSSEC being relevant 15 years ago but maybe not so much today – and while there are new requirements, like confidentiality, that were not maybe so important 15 years ago – made me think that maybe we have to start, before implementing, putting more protocols and your stuff onto the DNS tech. We have to think again whether the requirements for the entire system have changed and we are designing something that can meet them all. Otherwise, we’ll continue to be patching stuff here and there and maybe get to a result which is totally unmanageable.

Thank you.

JACQUES LATOUR: Thank you, Vittorio. Any questions?

UNIDENTIFIED MALE: Back here.

AFIFA ABBAS: Hello, everyone. Afifa from Bangladesh. My question was [on] confidentiality; to achieve of both of them, the confidentiality and integrity. You and Warren mentioned that we'd need both. But my question is, if we can achieve – there is no reason that we will use this – on the second option, DNS-over-TLS or HTTPS, there is no way to alter the traffic. So is it necessary that we use both? And if we do, is there any possibility to add the additional latency in the network?

VITTORIO BETROLA: Yeah. Let's say the official reply is that you have to do both because they are meant to do different things. So my [set of talks] was that maybe we can think a different way of seeing this question, but yeah, in the end, if you [inaudible] at having both, there are – DNS-over-HTTPS and DNS-over-TLS are more recent. There are lots of questions and concerns, especially in terms of policy, so you should maybe first be aware of all the problems before starting to implement. But in the end, they give you something that DNSSEC cannot give you. So I think you actually need both.

In terms of latency, it depends. One of the issues with DNS-over-HTTPS is that it seems to promote some [centerization] in which, other than using our ISP's local resolver, you will use a remote resolver, like [Code 8 or Code 9] in Google. Whether this creates more latency depends on your connectivity. Of course, your DNS queries have to go farther way, but this doesn't necessarily mean that they are lower. It depends on your connectivity.

JACQUES LATOUR: Barry?

BARRY LEIBA: It was a little hard to follow some of that because you were talking very fast, but I think what I got as the bottom-line is that you are proposing a hybrid mechanism where the recursive resolver verifies DNSSEC from the authoritative server and we use HTTPS to deal with the communication with the recursive resolver and we trust the recursive resolver so we don't need to verify the DNSSEC at that stage.

Is that what you're proposing?

VITTORIO BETROLA: Okay, I'm sorry. I think this is still – the entire [set of talks] is not to clear to me, so it's not really something I'm proposing. I'm rather thinking of different possibilities. I think I actually wanted to get feedback and understand whether they're interesting.

I'd say that this is also a model that could work. It does create the issue that you really need to trust your resolver, so it actually creates more policy issues on how do you choose the resolver. It also builds on the assumption that you only use one resolver that you choose, while there are people that actually would like to distribute queries to any number of resolvers. So I think it's very early to make any suggestions. But it's something that I think we could continue. Especially since we are now discussing the, let's say, policy issues with DNS-over-HTTPS and so on, maybe we could consider different scenarios before designing the solution for the next 15, 20 years.

JACQUES LATOUR: Warren?

WARREN KUMARI: So I suspect that a lot of this discussion will happen again in the IETF, etc., but Vittorio raised some sort of concerns – and we should definitely discuss them more – about the browsers forcing users to opt into this. Of course, Google is the largest company, so it takes a while for us to be able to see anything. But there's some rough, illustrative plans and desires that we're trying to adhere to, and those include things like making sure we don't surprise our users. The current plan under consideration is, if a user's got a resolver already configured, considering testing to see if that resolver that they're already using does DNS-over-HTTPS or DNS-over-TLS and opportunistically using it if it can. Also, there aren't plans to force a specific resolver without the user consent.

So that's somewhat going orthogonal to this, but I thought it was an important topic because a lot of people are really interested and excited in it at the moment. That's all.

JACQUES LATOUR: Wes?

UNIDENTIFIED MALE: Hold on.

JACQUES LATOUR: Oh. It was Wes first and … which one was first?

WES HARDAKER: Either way. My mic's on, so I win, right? Sorry. Interesting presentation and an interesting chain of thought. One of the things that I think felt a little bit misplaced is that you said that users will trust their browsers. The reality is that the users are trusting their browser vendors with a transitive trust to whoever is actually providing the resolution because, right now, the browser vendors are not actually the ones providing the resolution. They might have deals with Firefox, the most famous being Firefox and Cloudflare having a partnership, and we've notoriously had issues with users not being able to make informed decisions of trust. So that puts us in an interesting predicament because it also means that users actually trust their browsers more than any site they're going to because that partnership may actually be forging requests and sending to you to different places that you don't know about. You're trusting that browser more than the site that you potentially want to get to.

VITTORIO BETROLA: Yeah. Maybe before we get the other, I just wanted to clarify that this was not meant as browser bashing, actually. So I actually appreciate Google's approach to this. But it's true in the end. All this architecture, both in terms of governance [inaudible] of the root server system, ICANN and so on, is something which is something which is completely unknown to the user, while the browsers aren't. So if users have to pick someone who [to trust], I think browsers are more likely to be that.

JACQUES LATOUR: Yeah, I agree. We have more questions than answers right now.

STEVEN CARR: Hi there. Steven Carr for Infoblox. I kind of understand where all this technology is coming from, but the one thing we really need to make sure where that doesn't happens is that the browsers essentially take over. DNS is an underlying protocol. To do all of this in the browser, in my mind, is the wrong place to be doing it. Then there's still a whole host of other applications on the operating system that aren't going to be protected by this technology.

So to what Warren was saying, as long as the browsers have a mechanism for being able to determine if the operating system is already secured and to leverage that – but to just have the browsers in place overriding the security? That's something we need to be very careful that we don't fall into that trap because the users will then have a false sense of security that their browser may be secure but everything else that they're doing on their system is not.

JACQUES LATOUR: That's why we're talking about this now. Agree.

Warren, you want [inaudible]?

WARREN KUMARI: It seems that the writing is on the wall, that, increasingly, applications are going to be doing their own DNS resolution. Whether people think it's a good idea or not, it seems that that's the way that things are heading.

I don't think that's caused by DoH. I think that that's just, as applications change and some people start doing this, other people might decide it's a good idea. No value judgement on if it is or isn't.

So, as an example, for a long time now the Netflix app on phones and I think Rokus, etc., have been doing its own resolution already. It's very easy for things to do their own resolution. And, yes, this is a definite change to the architecture, and it's something that should be discussed and investigated.

But I don't think it's browsers versus operating systems. I think it's applications versus operating systems. So it's a much wider topic.

So, yeah, it should definitely be discussed, and there is, for people who are coming to the IETF, Stephane Bortzmeyer has organized a side meeting specifically around this sort of topic. And there are many threads on it.

VITTORIO BETROLA: Yeah. If I may add, I totally share your concerns since I'm one of the people on the IETF that is trying to prompt some discussion. So please join the discussion.

Another point I'd like to make that not all the applications be able to do all the resolution on their own on the other hand, so you do have problems with – if you move all the current security stuff which is being done at the DNS resolver level to the applications or to the end point, which is appealing to people that are against censorship, the problem is that maybe your smart refrigerator will not be able to run antivirus and antimalware solutions and actually provide that security. So there are some serious problems, even in terms of security, [inaudible] application resolution.

JACQUES LATOUR: That's it. Any more questions?

All right. Thank you, Vittorio. Next up is Duane Wessels from Verisign to talk about the KSK Rollover Post-Analysis.

DUANE WESSELS: All right. Thank you, Jacques. So I have some data I want to share with you that we're seeing after the KSK rollover. So this is the schedule of events that have happened to date. One is still to happen. But what I'm focusing here is on things that happened on October 11th, which is when the rollover itself happened – oh my goodness.

UNIDENTIFIED FEMALE: That didn't fix it.

DUANE WESSELS: And on January 11th, which is when the key was published with the revoke bit set. This data comes from a couple of different sources. Most of what I'll talk about comes from DNS query traffic to the root servers that Verisign operates, and I'll say a little bit about the RFC 8145 key tag signals to those same root servers.

Okay. So this bunch of graphs to show you are all graphs of queries for the root's DNS key. So these are .in DNSKEY queries. This graph shows a little bit of time before the rollover and immediately after the rollover. The Y axis here is millions of queries per day to A and J Root. So before the rollover, we were seeing about 10-15 million per day. After, we're seeing about 70-80 million per day. You can notice that this line goes up a little bit slowly over the course of a couple days, when TTLs timed out and what not.

These graphs show individual IP addresses. I include this just to show that individual addresses have different behaviors. For some of them, the levels went up and stayed up high. Some were high for a little bit and then went back down. So looking at individual addresses here was not particularly fruitful, just because they were all a little bit different.

So this next graph shows the changes at revocation. We include the rollover period as well, but you can see, at the time of revocation, we saw another significant increase. This shows about the first ten or so days after revocation.

Here's another month or so. You can see it continued to go up and up.

Here's the most recent data. So, a couple months after revocation, these two root servers are now seeing a billion queries per day for the root zone DNSKEY RR set. That's an increase of about 100 times prior to the rollover. That compromises about 6% of all the traffic that we see at these root servers. So this is pretty significant.

Obviously, to the extent that this would continue to increase, it's going to become more and more of a problem. So a few of us have started to do some outreach to the networks where we see these queries coming from, hoping to understand exactly what the causes are and eventually to get these fixed to stop this kind of traffic.

Here's a similar graph that shows, again, the behavior of individual addresses, just to demonstrate that there's a lot of diversity here.

So one question that has been asked about this, could this change be due to large responses carried over IPv6? Because the size of that response, the size of the DNSKEY response, is now 1425 bytes, which is a local maximum. It's never been this big before. And it's larger than the IPv6 minimum MTU size of 1280 bytes.

So I think that's a reasonable question to ask, but to me this really does not really look like that kind of behavior. It doesn't look like time out and retry behavior. We're seeing tens to hundreds of queries per second per address from certain sources. We see it over IPv4 and IPv6. To us, it looks very much like what's been called rollover-and-die. Maybe some of you remember rollover-and-die. This was a term coined by Roy Arends and Geoff Huston and – I'm forgetting who else; a couple of others. Ten years ago, there was an incident where RIPE was doing some key maintenance in its in-addr.arpa zones. This was before the root zone was signed, so people were using locally configured non-root trust anchors. When RIPE removed the old key from their zone, there was this huge spike. This graph is taken from their report. So to me, this feels very, very similar.

So now I want to show you some visualizations that explore this data a little bit further. I'm going to have to – oh, yeah. So, before we get to the visualizations, the idea here was to understand a little bit more about these individual sources and what kind of queries they're making. So we know that they're making obviously a lot of DNSKEY queries, but what other kind of queries did they make? Are they making only key queries or are they making lots of other ones?

So the idea was to put together scatter plots with one axis showing the count of DNSKEY queries and the other axis showing the count of all the other types of queries.

If you do that in a very simple way, you get something like this, which is not very useful because there are these significant outliers, with most of the data clustered around the origin. So the first thing you'd try is to make the axes logarithmic and you get something like this, which is better. But then you still have this problem where at the lower counts there's a lot of overlap and a lot of striping going on.

So what I did next was spread that out, add some randomness to the data, so that those points that are all coincident get spread out a little bit. So this is an improvement, but it's a little bit dense. So I made the dots smaller, which is good on one side but not so good on the other side, where they sort of disappear. So then I made the dots of variable size, proportional to their X-axis value, basically. So now you can see the whole thing.

Okay. So now hopefully you understand how these graphs and what you're looking at. You can see there's kind of diagonal line, a Y=X kind of line. Just to reiterate, each dot is here one IP address over every hour. Any dot that is below that line is making more key queries than non-key queries, and any dot above that diagonal line is the opposite. It's making more normal queries than key queries.

So if you think about what you would expect a well-behaving, normal recursive name server to look like, it would be in this band here, where it's making a very small number of key queries per [interval] level and lots more other types of queries.

All right. So I'm going to try to switch this and show – oops. This was Chrome, wasn't it? Darn it. I hit the wrong one. That's the wrong one. I don't want to—

UNIDENTIFIED FEMALE: [This one].

DUANE WESSELS: I know. That was the wrong graph, though.

UNIDENTIFIED FEMALE: [inaudible].

DUANE WESSELS: Yeah. I got to move this.

Okay. So here's this animation. This is for a ZSK rollover. So this is before the KSK roll. This line here represents the time – this is actually not a ZSK rollover. This is the time when the new ZSK is pre-published in the zone. I include this here to show a little bit what normal looks like, to show the animation works and whatnot. You can see lots of dots dancing around, but for the most part, it looks the same before and after the event.

But if you look closely, you can see one very interesting thing. It's easy to see down here. You see some green. I forgot to mention that green is v6 and purple is v4. You'll see green appear here. If you have really good eyes, you could actually see that, at every even interval, there's these bands of green. So what seems to be happening here is that, for v6, they're doing a query for the key. It's bigger than their MTU size. It's getting truncated, and then they're retrying. So, for v6, we see the counts on even intervals. They're seeing multiples of two queries per source IP address.

And then some other interesting things here. You can see some stuff moving around here and whatnot. Also, you'll note that, if you look right here, you'll see a pair of dots that appear. This seems to be a source that, every day at a certain time, some Chron job runs and it spins up and it does thousands and thousands of queries and then goes away. So it's a very brief spike of queries, but it happens at regular, daily intervals.

And you can see that, after the key is published, there's a pair of dots, whereas before, they're just on top of each other. So kind of interesting.

WARREN KUMARI: What is the [inaudible]? What's the big [standard]? Yeah, that one.

DUANE WESSELS: This one? Yeah, this is a good one to keep an eye on in the future graphs, too. This was a source. I believe it was in Korea, but I don't know much more about that.

All right. So this is for the rollover. Again, you can see here's our event when the rollover happens. That dot that we were just talking about you'll notice went down and then it went back up.

So I would assume they had trouble resolving. For a couple hours, they were done, and then they fixed and it came back.

WARREN KUMARI: [inaudible]

DUANE WESSELS: If you watch from the end and when it looped back to the beginning, you can see this area changes quite a bit. There's a lot going on here, where there's a lot of sources that became more active down here, where they're sending many more key queries. Very different from the start of this time period.

All right. Has anyone seen this enough?

UNIDENTIFIED MALE: Duane, can you explain this even behavior of queries of a lot of clients?

DUANE WESSELS: That I was talking about before? The green stuff?

UNIDEINTIFIED MALE: No, no. I mean almost the same number of queries of DNSKEYs and others.

DUANE WESSELS: Oh. Why there's this sort of strong line here? Yeah, I can't. I can't really explain that.

And I should also say that, again, if you think about the way we would expect a well-behaved recursive name server to work, it should only be seeing one DNKSEY query per TTL per day. So those would all be in this range. Anything here is already abnormal. Either it's not your typical caching name server – maybe it's a script or a tool that's doing probing or something – but it's not really what we think of as a well-behaved recursive name server if it's sending hundreds of key queries per day. Maybe these are a lot of things that – I don't know – are doing probing or measurements, but it seems like a lot of sources that are doing measurements. It's hard for me to believe there's so many.

UNIDENTIFIED MALE: And this is only for data for the IPs that are querying the DNSKEY, right? It's not all?

DUANE WESSELS: This is only for source IPs that send at least one DNKEY query, yeah. This is not all source IPs. That's right.

UNIDENTIFIED MALE: The key being requested – is that the key of .com or—

DUANE WESSELS: No, the root.

UNIDENTIFIED MALE: The root? Oh, okay.

DUANE WESSELS: Yeah.

UNIDENTIFIED MALE: Okay. Thanks.

DANIEL MIGAULT: I got another questions regarding the queries. When the query of the .arpa changed, there was this request-and-die thing. Do we understand what was the reason for this request-and-die?

DUANE WESSELS: Yeah. Let me talk about that in a minute. First we'll just cover this animation. So this animation covers the data of the period of revocation. So you can see that, at the start, it sort of looks the end of the rollover. We have a lot of clustering down here. At revocation, there's this very, very sudden change, where a lot of these whomp over to that section and they become even more active in their DNSKEY queries.

There's some really interesting patterns going on here, which, again, I don't have an explanation for. I don't understand exactly what's going on there. But—

[JACQUES LATOUR]: [inaudible]

DUANE WESSELS: Pardon me?

[JACQUES LATOUR]: It's a worm.

DUANE WESSELS: It's a worm?

[JACQUES LATOUR]: Yeah.

DUANE WESSELS: Yeah. It could be a little worm-like, but you can see it's a very significant change right at revocation.

So this data goes up until just a couple days after. This goes to January 15th or so. I don't have this for current data, which, again, we saw even a more significant increase within the last few weeks of this data. But this is just right after the revocation.

UNIDENTIFIED MALE: [inaudible] revocation [inaudible]

WARREN KUMARI: [inaudible]

UNIDENTIFIED MALE: How many IPs – [inaudible] – jump over? Is that order 100? Order 1,000?

DUANE WESSELS: I'd say order 1,000, yeah.

So I don't know how much time I have left, but I should keep going. We were on this one? [inaudible – okay. So in the slide deck, I included before and after images for those, since obviously the animation won't fit here and won't stay here. But you can look at that.

I guess before that I'll try to answer Daniel's question. You were asking what people knew about the rollover-and-die behavior. So that report did a very good analysis and found that – I believe they talked a lot about BIND software. Essentially, when it was doing DNSSEC validation and that validation failed because the keys are no longer present, it was very, very aggressive in retrying. It would retry every name server over both v4 and v6.

I believe it was even a little bit worse than that because, if, for example, the lookup was for example.com and, say, there were two name servers for example, the 13 for com and 13 for root – I believe it multiplied all those exponentially. So it was very, very aggressive in its retrying.

And it only negatively cached the failure for a very short period of time, like three seconds. So, essentially, almost every query to that name server resulted in tens or hundreds of key queries to the root.

Now, since then, current versions of BIND have gotten better. And I believe the report also talked about Unbound. I believe Unbound had similar behaviors. I think both of those software packages have improved, but I think are still not – well, anyway. I think there's still some issues there.

WARREN KUMARI: And I believe there's implementations who are largely doing what they believe the RFC said they should. It wasn't a coding bug. It was a, "This is what it sounds like we should do."

DUANE WESSELS: Yeah. Okay. So one thing we did was we took our list of IP addresses, making lots of key queries, and asked them for their version.bind. We got some answers. Whenever I look at this data, I'm always a little bit hesitant to have a lot of confidence in this data, but nonetheless, it think it's maybe instructive. There is a lot of BIND here. There's a lot of older versions of BIND here. We didn't get a great sample. We didn't get a lot of answers, but we got some. Some were open to our queries, and this is what they told us.

So I'll talk a little bit, just a couple slides, about the key tag signaling data. This is at the time of revocation. The green are sources that said they had both the old and the new key. The blue are ones that said they had only the new key. So, for example, you might say that, well, this very small number down here are ones that were manually configured to use only the new key and they mainly removed the old key or something like that.

This vertical line is of course the time of revocation, and you can see that it changed very quickly. Within an hour or so, half of these signalers had changed what they were reporting.

The red line here are the ones that said that they have only the old key. Their level is unchanged by the revocation, which is kind of what you'd expect. They are not really paying attention to RFC 5011, so they're unchanged.

There's some interesting going down here, where there's a small smattering of things that are reporting that they have the revoked 2010 key and the 2017 key, which is maybe something we didn't expect. Actually, I'm not sure the RFC is very clear on how to handle keys with revoke bit, but it seems weird that you would report that you have a revoked key in your trust anchor set.

Then there's even one that says, "I have only the revoked key in my trust anchor set," so that's kind of weird.

This is just the same data with a little bit longer time on the X axis here. So just to show that these things have also not gone to zero. They're hanging out in the range of 10% on each side.

Nah, I think that's what I just said.

The last graph I have in the slides is a little bit confusing, but the idea here was to combine these two data sets together and see, for those sources that are in this DNSKEY top talker set and that are providing signals, what we can we learn about them? I realize that you can't really read this legend, but the different colors represent different combinations of key tags that we see from them, some of them even being IP addresses that say different things at different times. “One time I have this set, and another time I have this set.” But just to show that there’s some diversity here.

The way you could interpret this graph is the lines that are in this area are addresses that we see more consistently over time. They’re a little bit more stable. The ones up here are more dynamic. We don’t see their reports consistently, day-to-day. They’re coming and going, so maybe they’re mobile devices or something like that.

So that’s the end of my presentation. We took some questions already. I can take some more if there’s time. I don’t really know.

JACQUES LATOUR: We’ll take questions after Wes, if that’s okay.

DUANE WESSELS: Okay.

JACQUES LATOUR: Or …

WES HARDAKER: [Yes].

JACQUES LATOUR: Yes? Okay. Thank you, Duane.

WES HARDAKER: [inaudible]. Thank you. All right. I'm Wes Hardaker from USC ISI, and I'm going to talk today about a week of data that we saw at our root server.

I have an addiction, and that's to create graphs that I don't understand. So that's what you're doing to see today because I want to spread that addiction. Previously, I've come and presented material where I had answers, and presenting an answer often takes 20 minutes. Presenting graphs that I don't understand should take less, which works because I'm also hungry.

So I mentioned the week's worth of data. I'm going to point out some sort of individual clients that are interesting. Then I'm actually going to talk about how we're releasing, basically, a week's worth of data for other people if you want to look at this same data set that's been anonymized in the way that we do for OARC. I'll come back to that at the end.

So this week of data is from the 9th to the 16th of January from this year. As Duane already mentioned, the revoke bit was set on January 11th. This is what the week of data looks like. It's very periodic, but if we graph it, you'll see that there's a number of spiky, odd dots at the top. I'll talk about that at the end of the presentation when I describe the week of data. But it goes up and down every day, like most traffic does.

If we graph just the DNSKEY queries – so, in most of these graphs, you'll see a vertical line, which is right where January 11th is – I'm not sure why there's a drop. So one of the things is that I did a lot of this work in the last week and I was completely swamped before that. So I have questions about the data accuracy in some of this. So I don't know why there would be a drop there. My gut feeling is that somehow the data is broken because why would people be querying less right after the revoke bit is set? I'm not sure.

But basically this is the number of keys queries per hour. You can see that there's sort of ramp, that the data on the right is slightly higher than the data on the left.

One of the things I also looked for was, okay, well, if they're asking for DNSKEY queries, what are they asking for? The vast majority of them were asking for the root, but I did find it interesting that there was a lot of other queries for other popular names, like net and arpa and things like that.

Typically, if you follow a resolver's behavior, it would be unusual that they would ask for keys for other things in general. I do love the fact that they were asking dlv.isc.org, which I think has been shut down for two years now? Something like that. Yeah. And it's a lookaside validation, which means you shouldn't be asking for its key in the first place. It should be pre-configured. So I don't know what's going on there.

So when I broke, going back to this graph, this list of DNSKEY queries down into individual addresses and then looked at who the top talkers are, you can clearly see that there is a large spike on January 11th, where each of those colors is basically a different address. You can see that some addresses are suddenly ramping up rather quickly and asking for DNSKEYs a whole lot more.

We're going to look next into the more interesting of each of those colors. So this is top talker #41. I call this, "Is it here yet? Is it here yet?" This is probably somebody was asking for the DNSKEY repeatedly until it changed, and then decided, "Well, that was boring," and went way. So this is sort of the inverse of the problem. This is somebody asking for a whole lot in the beginning, and then there's two little dots in the far right-hand corner that you can see that maybe they went back to normal operational practice. I don't know.

UNIDENTIFIED MALE: Probing?

WES HARDAKER: Probing? Yeah. So quite possibly some monitoring system that was waiting for the event.

#16 in terms of top talking had this ramp-up. They were fairly low in the beginning. They're all down near the zeros. All of these are requests per five minutes, by the way. Then they ramped up and they stayed steady after that.

This is an interesting one that's probably related to probing, too. I originally titled this "Early Issues," because, before I drew the line on the graph, I thought, "Oh, they had issues for an hour and then fixed their problem and went away." Then I drew the line on the graph and went, "Oh, they're talking before the event. Interesting."

Some people had fairly late issues. I don't understand because typically a resolver should be querying in the first two days. They should get the new key. They would cache it for two days according to the TTL, and eventually they ask. That's where you would expect the problem to occur.

No. Not in this case. And note – by the way, we're going to come back to this – that this one has a lot of high points and a lot of low points all in the same time period, which is interesting because that would indicate that they're asking a lot in five minutes and then all of a sudden they're not asking that much in five minutes, in a different five-minute period.

I don't know. #76 was asking a lot in the beginning and then went silent for two or three days, right at the revoke bit set, and then came back and started asking a lot. I did mention beforehand that I can't explain these graphs. I can only show them to you.

There is a number of cases where people got worse over time in stair-step kinds of patterns, and this is definitely one. There is a large population of NATs out in the world that we're well aware of that multiple resolvers were forwarding through a single address. You could see things like this.

But I will note, again, that some of those stair steps are well after two days. Why are they increasing long after TTL periods? I'm not sure.

Sometimes things got slightly better. It could be things shutting off. It could be things actually figuring out a problem, again, if it's behind a NAT or something like that and they started stepping downward.

This is the number one top talker. They queries on the order of up to 15,000 every five minutes and in a very interesting pattern of spikes once every day or so. The only thing they're asking for in this graph is the DNSKEY. So when I was looking at, well, what else did they do? This is it. All they're asking for is the key. All of those. So there are no other bullet points, in other words.

This is my favorite, and not just because it's 007. But it went along for a long time, then dropped. Still not reasonable levels, but dropped and then went back up again. I can't explain that at all.

So one of the issues of trying to explain this is that we have very little visibility. The 8145 data was actually almost easier to look into. With the actual looking at the real requests, almost all of their queries in these cases are for DNSKEY data. You have very little other visibility into what's going on.

So one of the things I decided – and I actually did this two days ago – is I wanted to see, well, what if we look at the rates that they are querying at both their minimum rate and their maximum rate? Because I mentioned before that some of them were querying really quickly and then were not querying very quickly. So this is the minimum and maximum rate per hour.

So you can see that there's that green dot on the far upper left that's gigantic and is way up high. And then there's a minimum rate. So every single dot has a corresponding green and purple dot. I don't know if any of them are covered up, so I apologize for that, that I didn't try and determine a vertical line or anything. There's many ways I want to improve these graphs.

But anyway, the point being is that there's a significant gap between the minimum and the maximum, which means that there's an awful lot of machines that are querying for a lot and then not querying very frequently.

If we zoom in a little bit to get rid of some of the outliers, we see some interesting patterns. There's definitely a large number of machines. These are all sorted by the maximum rate, so the things querying the most are on the far left-hand side, and then the minimums are just plotted below wherever the maximum fell. So there's basically the top 500 talkers along the X axis and then …

Now, this graph I did start, and by the time the slides were due, I wasn't really finished analyzing all the data. So I did notice that there's some interesting trends, like there's a sharp peak on the left and then there's sort of an elbow in the middle, where it kind of ropes down. That kind of goes away when the graph actually got finally finished in the final one. But there's definitely some top talkers on the far left, and then there's a sort of ramping down behavior.

But I want to draw attention to the minimum dots that occur underneath. There's this pattern of lots of talking and then these random smatterings of they're asking less than that. And I can't explain that. In fact, because I questioned data for this because I know there was a loading issue in the beginning half of the day, these graphs are all from the last half of the day because I wanted to make sure – when I originally plotted, I thought, "Oh, this has got to be an issue with the data," and then I knew the last half of the day was okay, so I replotted it and was like, "It looks the same." I don't get it.

So that's the last of those graphs. Again, I don't have answers for why there's this on-off behavior. I did reach out to one other university looking at where the data was coming from that had eight machines that were observing this behavior, and they seemed to go silent at night. They're still on because I can ping them. I did what Warren suggested I do, which is go ping them. They're still on, but they only send these requests during the day. And they send very little else, too. I looked at some of the other names, and it's like … So fortunately I tried to reach through some contents and I wrote their IT department two days, hoping I'd have this answer by today and I could give you a great resolution, but no. They didn't write back, so then I just wrote through a colleague's contacts in the last hour and I don't know. We'll see if they write back.

But anyway, as said, all of this came from a week-long data set, and we wanted to release this data set so that others could look at it, too. So on the bottom there is the Impact Cyber Trust Program, which is a way for researchers to release data sets. So we're doing through there. There are qualifications. You have to be doing it for research, not commercial purposes, and stuff like that. I will talk to OARC and see if they want the data set as well. I have not chatted with them. I suspect that they might. So it might be available through that path as well.

But basically it's the week that I talked about. It's January 9th through the 16th, and it has a number of interesting components to it, not just the fact that it's centered around the KSK revoke bit. But if you look at that bottom graph – the top graph is small but it's basically the first week; the bottom graph is about a day – there are spikes in the data for the number of packets that we receive. So these are zoomed-in graphs of our monitoring system. The lower baseline is our normal base rate traffic. Those bumps are all somebody querying from Amazon IP addresses that are two to three times larger than our regular query traffic. We don't know why they're doing this. We've talked to some other root operators and we seem to be a unique – they're targeting us. I need to talk to more of them because not everybody has looked at these. But I don't know what's going on there.

The queries that they're sending are just garbage. They look just like alphanumeric garbage. So if anybody wants to dive into this data, please talk to me and we can let you look at it.

I've done strange things. My first thought was Base64-related stuff, and Warren had the same thought. We talked it about. So I ran Base64 through strings and looked at it. It looks like there might be DNS names somehow encoded in Base64 bit. It's not a DNS packet. I don't get it.

I haven't had enough time to really dive into it, but I'd love somebody else to really dive into it so they can tell me if they could attribute this to the Amazon owner that is actually sending all this traffic. I'd love to turn it off because it's been going on since November. On a weekly pattern, these bumps occur. Reaching out to Amazon, they went, "We don't know who it is." How do you not know that? I don't know. And we reached out through two channels in Amazon and failed to turn them off. At this point, I'm interested, so I don't really care. I really want to figure out what's going on.

So any questions? Then I know we're going to go into question period. I have a lot of questions. I don't have answers, as I mentioned. So if there's any questions I can answer, I'm here. Otherwise, you can all just expound and talk about what you think is going on. So that's it.

JACQUES LATOUR: Thank you, Wes. It looks the data exfiltration via the DNS. Can you assemble all of it together and then we can … [oh].

UNIDENTIFIED MALE: [inaudible] the root servers.

UNIDENTIFIED MALE: [Sounds like you've volunteered].

JACQUES LATOUR: Well, somebody would have …

WES HARDAKER: Somebody fucked up the [inaudible], yeah.

JACQUES LATOUR: Any questions?

WES HARDAKER: Yeah, one of the ironies was that there was discussion about whether OARC should collect data on the 11th and everybody said, "No. It's going to be boring." Whoops.

JACQUES LATOUR: Yes. So we got ten minutes for questions, I guess for the entire panel. So …

Going once …

UNIDENTIFIED MALE: Oh.

[RUSS MUNDY]: Oh, yes. Thank you.

EDUARDO DUARTE: Eduardo Duarte for Duane. You talked about, on the first animation, IPv6 behavior, but you didn't talk on the second. They seem to be opposed?

DUANE WESSELS: Yeah. So the KSK events – so the rollover and the revocation – are all right adjacent to the ZSK events. So, throughout all those times, the – are you saying you notice that the striping was in all of the graphs?

EDUARDO DUARTE: Yeah, but on the second, it was before [e], and then after [it wasn't].

DUANE WESSELS: Yeah. So it really depends on the size of the DNSKEY response, and it depends on how many records there are in there. So, at ZSK rollover, we're adding signatures and, for a KSK revocation, we're adding a new key. So any time that the message size gets above that 1280 is when we saw that splitting of the IPv6 traffic.

Does that make sense?

EDUARDO DUARTE: Yeah.

UNIDENTIFIED FEMALE: [Wes, I have one]. This question is for Wes from Geoff Huston. B Root truncates its responses at 1,280 bytes and both four and six. To what extent is what you are seeing an artifact of this truncation of UDP responses and the subsequent issues with TCP?

WES HARDAKER: Thanks, Geoff. Can you clarify? We had this discussion a while ago, and I don't believe that – that's statement you're pulling is from two or three years ago is not current according to our current operational practice. Have you measured that recently?

Nonetheless, I can answer it. I have not looked to see if there was a big shift between UDP to TCP. That is something I should definitely graph, but as I said, I have only been dong this for a couple of days.

UNIDENTIFIED FEMALE: Geoff has said he has not measured it recently.

WES HARDAKER: So we talked about this six months or a year ago, and when you made that similar comment at another forum, I said, "No, no, no. That's old data. You have not measured that recently." I'd love it if you would so that you can verify that I've actually fixed that problem. But before the DNSKEY rollover time period, we made sure to change that. So it shouldn't cause an issue.

UNIDENTIFIED FEMALE: Geoff Huston also states, "A and J also truncate IPv6 at 1,280."

JACQUES LATOUR: So when is the next phase for delete?

WES HARDAKER: The next phase for what?

JACQUES LATOUR: For removing the KSK.

DUANE WESSELS: On March 22nd, that'll be unpublished from the zone.

JACQUES LATOUR: So should we plan to do a Doodle?

WARREN KUMARI: It'll be fine.

JACQUES LATOUR: It'll be fine?

WES HARDAKER: Yeah, I should have mentioned, by the way, that these graphs, the green and purple ones, I did on March 7th. So this is not from that week of data. These are actually later because I wanted to see if the problem had gone away. And of course it hasn't.

JACQUES LATOUR: Fred?

FREDERICO NEVES: It's Frederic Neves for Duane. Have you reached out to any one of those top talkers or …

DUANE WESSELS: Yeah. Not myself, but another root operator is in contact with the top AS sources. We don't have an explanation yet, but they're looking at it and hopefully will – I would love to have an explanation of what's really going on.

Usually, in my experience, when you do outreach like this, it's more likely that it just sort of stops and you never get an explanation. But I would love to really know what's going on here.

FREDERICO NEVES: Anyway, the graphs are great, guys.

DUANE WESSELS: And I also will say that it is my hope – if this level of traffic continues, I think we're going to have to be more aggressive about reaching out to people. So I guess keep an eye out for maybe announcement from us for help in tracking down ASes and sources that are behaving like this. We would love to have the community's help in finding and understanding what's really going on here.

RUSS MUNDY: Duane, question for you. Have you looked at projecting forward what some of these traffic growths might do over six months, a year, if they in fact did continue?

DUANE WESSELS: I haven't, to be honest, but it could be a little bit terrifying. Like I said at the start, by my calculations, these DNSKEY queries are already 6% of our total traffic. So that's pretty noteworthy.

JACQUES LATOUR: All right. Any more questions?

[That's good dressing]. All right. [Thanks].

WES HARDAKER: One quick data point because somebody just wrote me back about stuff.

UNIDENTIFIED MALE: Woo!

WES HARDAKER: Well, no. I wish. They wrote back with the fact that a lot of these coming from even DigitalOcean, Amazon, AT&T, and NTT, generically – and [NetBox]. That's unfortunately not helpful because those are rotating things a lot of the time.

JACQUES LATOUR: All right. Thank you, Wes.

[Kathy], can you provide thus the instructions for lunch?

[KATHY]: Yes. Lunch is actually just right across the hall, so you don't have to go very far. Just make sure you have your ticket and be back here at 1:30.

[END OF TRANSCRIPTION]