MARRAKECH – DNSSEC WorkShop Part I
Monday, June 24, 2019 – 09:00 to 10:15 WET
ICANN65 | Marrakech, Morocco

RUSS MUNDY: Good morning everyone, I think it's time for the next DNSSEC workshop to begin. I'm Russ Mundy and Jacques Latour and I are from the program committee, our family of folks that put this together for you each meeting. I want to thank everyone for attending, and point out that we like to have these sessions be vertebra informal. Questions are welcome pretty much at any time, particularly today. So, today's session is one half day, the second half of the day will be the tech day program. And so all of that will take place in this room and we do have lunch that will be served in this room, which is a wonderful treat for us, we don't have to go anywhere. I think that at this point I would like to turn it over to Jacques, who is actually going to do our first official intro here.

JACQUES LATOUR: Thank you, welcome. So, this is the program committee for the DNSSEC workshop. We meet once a week and we build the program for the next ICANN meeting. We do a Lessons Learned on the previous meeting and then we try to make the next one better. We put together a call for presentation that is based on

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

actual technology development, so we try to make it as relevant as possible for the next meeting. It's every week we meet to build this program. The most important thing about the DNSSEC workshop is the lunch so we have sponsors, many of them. We have a budget to have a lunch now, and so they sponsored lunch. Like Russ said, it's going to be in the back today. There is a lunch coupon on your desk. If you're in the room I guess you don't need a coupon. You have to go out and go back in? No? We'll figure it out. Thank you to the sponsors.

The DNSSEC Workshop is a collaboration between the SSAC, the ICANN Security and Stability Advisory Committee and the Internet Society. So we work with ISOC to build this program. So, today we're going to have a morning workshop, so I'll do the counts. Every ICANN meeting we show you all the relevant activity between the last meeting and today in terms of DNSSEC with TLDs and then we go through the counts and validation, and all that. And then Tim April is going to talk about the DNS Transparency Project,and then Wes Hardaker is going to talk about, you should know after this the difference between SMTP/DANE and MTA-STS. I'm going to learn something. Then we have a coffee break and then in the afternoon we have new presenters. Tim April is going to talk about KSK Roll and Wes again is going to talk about DANE development and deployment. And then we have just before lunch, Russ is going to finish with

how can we help the DNSSEC and what you can do, and then we have the best and the greatest DNSSEC quiz with 11 questions this time, instead of 10, so get ready.

So, Counts. Last session we had some issues with our numbers. This session we might have issues again, we'll find out. So global validation for DNSSEC now is at 19%, which is good, because we had a bit of a dip and then after the KSK Roll, we're now back up. So, hopefully people will turn it on more and their resolver globally. So that's obviously the next step, try to get more and more people to validate. So, that's a highlight of who validates by region, by subregion, and who uses Google DNS to do the DNSSEC validation. So you can see some countries are more aligned toward Google than ISPs signing on their own. For example, Micronesia, they do 5% of their own validation with their Is present and then 61 with Google. So those are based according to the APNIC stats. At least we are validating DSNSSEC there. There are some regions that we need to work on, so that's an ongoing challenge that we're seen. So those were the count. The last session we had issues with getting the right numbers.

Now we're looking at deployment by percentage of signed zone. Do we have the right number again? Yeah, we're good. Everybody we're good with what you have there? Fred, do you

have more than 30? Validation? So it shows Brazil 25%, roughly, okay? Good. So, we're making progress, we still have a lot of work to do to get signed zone and signed top level domains. And then what we do, so Dan York at ISOC tracks the deployment of TLDs. The ones that are going through the different stages. So experimental is they're playing with DNSSEC, announce means they made a commitment to someone that we've detected, that they want to sign their zone. Partial is they're working on the architecture, it might be they might have keys, DS in the root is partial too.

And then operational means they're accepting DNSSEC material from registrars. And I think this is the biggest jump we've had in a couple years. We had 6 changes between the last meeting and this one, so I'll go through that; .DZ got signed in April, which is a good step. In AP-TLD region, Kuwait is signed. In Europe ccTLD, we got three new TLDs, Moldova, Slovakia, Monaco that got signed. Then in Latin America we got two, Guyana and Anguilla. AI is experimental, except that one, we should remove Anguilla because I talked to the TLD Operator last week and he said that he's passing on setting up the DNSSEC for .AI. So, Tim and I were kind of trying to work with them to sign their top level domain, so maybe if you run stuff on AI and you want DNSSEC, we need some ways to convince the operator to sign their TLD. But so far the discussion we've had is they wanted to sign, and then they

decided to remove, so we'll have to update our slides accordingly. And then North America, we're at the same stage.

The maps, we publish them on a monthly basis. You can subscribe to get them and you get a bunch of maps and all the Excel spreadsheet and all the stats for the global deployment of DNSSEC. Like I said, this is updated based on discussion and AI is a good example of a TLD that decided to go experimental and now they're going back to not signed. And then if you have time you can go to the DNSSEC history project, that's the URL, and then you can review that, if you have any pertinent information that's not already on the website, you can update the site there or provide comment to Dan and update this or contribute. And that's about it. Any questions? Okay, so next up is Tim with the DNS Transparency Project.

TIM APRIL: I'm Tim April, I'm from Akamai, I'm also on the SSAC. For this I'm speaking as part of the group of people that's working on this project. The basic idea is, how many people here were at the high interest topic at Kobe that was talking about the DNS hijacks that were happening in the domain system? Okay, only a few hands. As a result, there were two major attacks that happened at the end of last year and the beginning of this year. They went by the name of Sea Turtle DNS Espionage, depending

on which papers you were reading. There was believed to be a nation state actor that was breaking into registrars and registries and even some DNS operators to change DNS records to point specific domains away from their expected authorities towards some other authority for very short periods of time. As a result of the short time window some monitoring tools wouldn't pick up on those sorts of hijacks so a few people who were involved in the response, including me and a couple other people from similar organizations got together and started talking about what can we do to help this fix this sort of thing in the future. So we started with the DNS Transparency Project, basically an Open Push Update system for DNS changes at the registry level.

The mission that we've so far decided on is to try and make the changes that go into the DNS visible to anyone who is interested and as close to real time as possible, where you would see the change to the DNS as it's being made, not just whenever you query it. The problem is right now, whenever you want to inspect the DNS you have to pull it, so you have to do a dig from your own machine, you have to set up some sort of [inaudible] job. There are some services out there that will monitor your DNS every minute or so, but if the changes are very quick and precise, some of those updates can go unnoticed, specifically

the short-term changes that may only be in the registry for a handful of seconds.

Just a quick overview of how the system works. The registrant will request a change either through a reseller or directly to the registrar. Then the registrar gets those changes, they will do their business logic or whatever they need to push the changes up to the registry and then that will go to their gTLD name servers. Then a resolver will go through and query each of those names servers and keep that information for whatever the TTL is, ideally. So the problem here is if the registrant credentials or the registrar or registry is compromised, they can push a change to the TLD name servers and the resolvers can go and get that data in real time, and they'll cache it for however long the TTL is. So, if you've got a 2-day TTL on your registry, any resolver that polls in the time that update was invalid, well then cash that for two days without you having any ability to go and inspect that resolver's cache. There are some ways to do cache flushes, but that requires that you know that you were compromised in the first place.

As I was discussing, some people build monitoring tools that will go and poll either the registrar using whatever API you have for the registrar. You can also have polling that just queries the name servers that are involved. In some cases you can also poll

resolvers, but you would have to poll every resolver on the planet, and that can be kind of flaky and not fully transparent to the end user because like I was mentioning, there are some update issues, the timing of updates and polling matter a lot. So our proposal is to create essentially a pub sub system where registries can send data and we can get push notification, switching it from pull to push, where updates come in a realtime stream, they get processed by some system and then shipped out to anyone who is interested receiving updates on that particular change. So this is kind of an i-chart, it got a little gangly towards the end of it.

But the basic parts in the center of this picture are the same as before, where the registrant pushes to the reseller or the registrar, it goes to the registry into the name servers. In phase 1 of this project our plan is to try and set up a system so the DNS Transparency Project, the green circle over on the far right side of the picture, builds a system that takes push based, so essentially either replicate either the zone file push to the name servers or even a mirror of some selective EPP commands into this DNS Transparency Project, where it will then build the monitoring system that receives those push notifications and pushes alerts either over email, SMS, just a generic API call to some endpoint, that registrants or even things like MSSPs, the security providers that some companies pay for, can subscribe

to these updates in near real-time and have an idea that something has changes, and possibly mitigate it if we get the right data before it hits the gTLD authoritative name servers for that zone.

In Phase 2 we're hoping to implement ways of integrating with registry back end providers and also with registrars if they're interested in pushing data to us, as well, so that you can double check, essentially you could tell if something were to be compromised, you can tell whether it was the registry, registrar or the registrant credentials that had been compromised. The reason was picked the name DNS Transparency was to try to model off the certificate transparency model where, if anyone is not familiar with that, when you sign a certificate using the web PKI now, most of the CAs will push information about what certificates they signed to a certificate transparency log where users can go and look through the log and find any certificates that were signed for domain if they're interested using these logs.

There are some cases where monitoring tools are built to go and pull that sort of data, specifically in the attacks in January and December, we used certificate transparency logs to find out what certificates were issues for some of the compromised domains. But as we were discussing this at the onset of this

project, we realized that certificate transparency logs are great, but they aren't consumable by the end user in a meaningful way unless you have the resources to go and pull that data in real time. That's why we're talking more about building a system the has the ability to push updates to the end users through either emails, SMS, or something like that, rather than having each end user who is interested go and build a system that will monitor the entire loge of data.

So, as I was saying, the input to the system we expect to be, we've been talking with a bunch of registries where they're trying to work out the details of setting up a system where we will receive their data and hopefully then process it and create the two goals that we're working towards now are a raw feed of domain changes that can be consumed by security researches, threat monitoring companies, things like that, and then filtered updates. So the raw feed is something similar to the currently certificate transparency log. The next step down is the filtered updates that will go to end users that don't want to build a full infrastructure to consume this data. We're in the process right now of trying to set up an entity to handle this. We're trying to create an open organization that can be as transparent and possible, so we're not trying to put it under an existing company.

So, we're working on getting through all the legal paperwork for that and the nonprofit status and everything like that. And then we're also starting the process to build a proof of concept with a couple registry partners that we've been talking with over the last few months, so we can guild something that we can get users to go and play with to see if it's even going to be a useful thing to keep pushing on. We believe it is so far.

So, we're looking for people that are interested in providing us data. So if you're a registry, if you're a registrar, and are interested in hearing updates about Phase 2, I'm here all week if you want to come talk to me. We're particularly interested in CCTLDs which were heavily targeted in these recent attacks. And then if you're a registrant or end user and you're interested to hear updates when we get to the point of actually having the concept up and running, there's a contact email at the end of this presentation.

And then there is also the help of if you're a company or group that may be interested in consuming some of this data, once we get the system up and running and the entity founded, we would be very interested to talk about donations of either end kind or monetary to help build the system, but we're still dealing with all the legal paperwork. If you would like to get in contact with us, there is the information at dnstransparency.org, this is still

very early stages right now, so the website hasn't been put up yet; we have a draft of it, but it's not live as of yet. I think that's all I actually have.

UNKNOWN SPEAKER: So, I might have missed it in the presentation, but is this just going to be a stream of updates or are you actually going to do something like put it in a certificate transparency type, blockchain log?

TIM APRIL: We're starting with the push based process, we're trying to figure out how to make it an auditable append only log in a reasonable way, because we don't have the full history of everything at this point and we're not going to have a competition data set at the beginning.

UNKNOWN SPEAKER: Yeah, I mean you could stop feeding it into a certificate transparency thing at this point in time.

TIM APRIL: Yeah, that is of interest. If you want to send email to that address saying that, or to me, and I will add it to our list.

UNKNOWN SPEAKER: For those of us that might be interested in this, is it likely that if we talk to you nicely we could get enough access to start building tools? Of is it still too early for that?

TIM APRIL: We don't have any data feeds yet. We've only got verbal agreements from a couple different organizations to actually do something.

UNKNOWN SPEAKER: But like are you thinking about like an API?

TIM APRIL: Yep.

UNKNOWN SPEAKER: Okay.

TIM APRIL: We're starting to design the API, we expect it to be built on some sort of system where we're working on defining all the data formats and things like that, but we hope to work with at least a

few consumers of it very early on to make sure that we're building the right thing.

UNKNOWN SPEAKER: So, it sounds like I should nag you again next time we meet.

TIM APRIL: Yep.

UNKNOWN SPEAKER: Okay.

MICHAEL CASTOVAL: Michael Castoval. So, a couple points I want to bring up here. One major problem is you're not adding any level of security or transparency to the recursive resolver level, which is the most vulnerable, because the recursive resolver can easily feed false information. I've been doing some tracking and studying of this. Is there any plans to like, for example, in certificate transparency, X509 certificates can have an integrated hash built into the certificate so the end note can be found. Are you looking at extending the DNS protocol or adding an EDNS extension so DNS transparency information can be checked by

the end user and verified to be correct, so that the recursive resolver is less resistant to tampering?

TIM APRIL: I believe that to be out of scope of the current project where we're mostly focused on handling the resolution at the authority level. At this point we're trying to get this thing off the ground to the point where we're not editing the DNS part of this is probably going to be probably out of scope for right now.

MICHAEL CASTOVAL: It just occurs to be me that without bringing it to the DNS protocol level, you essentially have this giant repository of database of information that has to be checked by an entirely different process that sort of seems to limit some of the utility. That's just my 2 cents, perhaps when I see the documentation more in depth and the design, I'll change my mind, but that's my 2 cents.

TIM APRIL: My initial thought is that DNSSEC is probably the way to get that sort of resolution changing.

MICHAEL CASTOVAL: DNSSEC doesn't solve the problem, the DNSSEC information doesn't travel to the last mile. The recursive resolver sets single byte and can lie all it wants if DNSSEC information is valid or invalid. My home ISP just sets it to true, no matter what you point it at, as I've discovered in my research. I've been doing quite a bit of research on recursive resolver behavior. So I would like to see more security in that field, if possible.

UNKNOWN SPEAKER: I got a question. As a registry what kind of information would you expect us to send to the transparency project?

TIM APRIL: The current things we're looking for, we're looking for everything except for contact information.

UNKNOWN SPEAKER: Not everything. We can't sent everything.

TIM APRIL: As much as people are willing to be able to send. The most important information for us is the NSSEC, whatever would be in the zone file and is near real time as possible and another thing we're talking about is potentially a flag of when contact

information has been updated. We want to stay as far away from any contact information as possible, and this updates renewal sorts of information, DS key changes, and any other records that may be get put into the zone.

UNKNOWN SPEAKER: So it sounds like this is going to be somewhat annoying for registries to do and a lot of registries already don't like doing things like publishing zone files or submitting stuff to CCDS, et cetera, because it has a cost and because they think it's proprietary information. It seems like if somebody were to monitor this feed they could see when a new domain is created which potentially leads to spam, et cetera, things like that. How interested have registries been so far in providing this sort of information? Or is it more, yeah, it sounds interesting, we'll get back to you?

TIM APRIL: The conversation we had so far I would say is very positive. We don't have anything, like there is no contracts or anything like that created so far. The new creation spam is a concern that we're trying to work out, specifically for creates or deletes of domains, we're talking about a significant delay to unvetted users. That still is a key part we're trying to figure out. We would

like it not to be the 24 hour delay that CCDS is, but if that's where end up, that's about all we can do at this point.

MICHAEL CASTOVAL: Michael Castoval again. Do you intend to support information being pushed to the Transparency Project via AXFR or IXFR, so it can directly be integrated? Basically I can set my bind server and have it dump all its records right into the transparency level, from a second level or higher, if I want my entire zone file to be transparent?

TIM APRIL: We're considering something like a DNS notify and then we would poll the zone through some method. We're trying to make it as easy for a registry to integrate with us as possible, where either they notify us through an API...

MICHAEL CASTOVAL: I'm thinking more of the case where essentially the types of DNS registries that give you a free subdomain which would also benefit from this type of work, hence why I would like to see it go, if possible, be able to handle information coming from a level above the registrars and registries because sometimes you

have like no IP.com or DNS with quite a lot of users, that could also be vulnerable to hijacking.

TIM APRIL: That's all been currently deferred to a later phase of this. We're focusing on registry at this point and we've talked about going further in the future.

WARREN KAMARI: Warren Kamari again. So, sort of related to the previous question, there is also a DNS exchange project which is being discussed/starting up, where people can just contribute their entire zone file and changes, and then anybody else can slave them. That seems as though it accomplishes a large amount of the same sort of thing and also allows people to slave stuff, but that's still very much in startup phase.

RAMOHAN: Thanks, Ramohan from Aphilius. What happens to the data once it's collected?

TIM APRIL: We're unclear at this point. We expect the data to come into the system, be processed, and then be put into the pub sub system

and either consumed by either the process that is notifying the interested parties in the domain or being sent as a raw feed to wherever, we're not intending to store it for terribly long, we expect to keep the last state of the domain around so that we can determine whether or not something has actually changed or if it is just a re-pushing the same information. At least one of the registries we've talked to, and I can't remember which one, has expressed some concern about just keeping the data indefinitely, and we understand that.

RAMOHAN: Yeah, I'd say as a registry operator, I'd say participation in something like this would be quite dependent upon having clarity on the data collection, data storage, data retention, et cetera, all of those practices. If those aren't defined and those aren't clear, I don't know that the various legal folks are going to sit there and say yeah, you can send your data across. So, I suggest to get this off the ground, that part of it has to be locked down.

TIM APRIL: Yeah, we've been talking with some registries about what their needs are in that case, so that we can build a system that aligns with as many as possible.

UNKNOWN SPEAKER: I got one, Kim if you can go back to Slide 7. So, the output of the Transparency Project goes to security providers, and then the last line is from them to registrant. And I'm thinking as a registry, if I do have domains that have registry lock and somehow they get changed and it makes its way back to them, there is value in there. But the average registrant if they get notified from someone saying something happened, they would not even know what to do, I think.

TIM APRIL: Yeah, I did forget to put the line in here where the registrar or registry would be interested in receiving this data as well, just as a way to check themselves again to make sure, if I'm a registrar and I requested unlock of a domain, I'm then going to wait to see that notification come from the DNS Transparency Project to make sure that the unlock happened, and then if I receive an unlock that I didn't request, I'm going to pick up the phone very shortly thereafter and call the registry and say what just happened.

UNKNOWN SPEAKER: So is there plans, I think somebody asked the question about the recursive somewhere, like passive DNS data, to bring that in?

TIM APRIL: Not currently on the roadmap, but we can talk about that.

UNKNOWN SPEAKER: Because if we operate the name servers and we know what change is in there, spondylosis if you notify us something changed we already know, I'm thinking.

TIM APRIL: I'll add it to the list.

MICHAEL CASTOVAL: Michael Castoval again, I'm sorry if I wound like a broken record. If you're not intending to store the full information history of the DNS change log, you're not going to be able to use a Meckel tree design like certificate transparency. That means it's going to be impossible to independently verify the consistency of data stored by DNS transparency. How do you intend to address this issue, then, because essentially you're going to have a third party data store that cannot be independently validated for correctness because without Meckel tree which needs to keep a full history of all the changes?

TIM APRIL: The goal that we're currently targeting is to just notify someone of any change to the DNS that is related to their zone. The full history of blockchain sort of approach mentioned earlier, we've considered it and ruled it out of scope for the current approach. We could add it later, but that goes to Ram's point of the data retention issues, and things like that, that are of some concern, but currently unaddressed.

RUSS MUNDY: So, I don't see anymore in the queue or questions online. So, I wanted to thank Tim very much for this presentation and it's one of the type of presentations that audiences and participants in previous workshops have asked to see, new and some innovating things that are going on, which is exactly want this is.

The next presentation is similar, but a little closer tied to DNSSEC as it is being used today in the internet. Wes Hardaker is the presenter and Viktor Dukhovni are the ones credited. So, Wes, over to you.

WES HARDAKER: Thanks. My name is Wes Hardaker I'm with the University of Southern California's Informational Sciences Institute, and I'm going to talk today about SMTP security options. In the past I have given talks about SMTP with respect to DANE and how to

secure TLS connections using SMTP with DNS protected DANE records. The beginning of this will be a little bit of an overweight about previous talks about that, so I'll recap some of that, and about how SMTP history works. And then today I think it's the first time in the DNS workshop we'll talk about MTA-STS which actually doesn't require DNSSEC and we'll get back to that and we'll fracture or deformity a comparison at the end.

So, first off, can we zoom out just a little bit, that's not quite the full slide, it's close to the full slide, there we go. With respect to SMTP security, with email security, how the user interacts with a mail server is actually generally fairly secure. When a user, say on the bottom left, is going to be actually sending a mail message, it's going to communicate with its mail server and it typically does this over a TLS protected connection. That's why when you configure your mail server you need to configure both the incoming and the outgoing server properties, that includes the TLS protection and how to talk to your mail server. On the receiving side typically you would use something like IMAP or secured POP in order to receive the mail to your mail client.

However, between the two mail servers, until somewhat recently, there is sort of this miracle approach where the mail servers were communicating in the clear, man in the middle attacks are easily possible through routing changes or DNS

spoofing attacks, or things like that. And so it was very hard for mail servers to figure out how to create a secure connection so that when your ISP needs to transmit one of your mail messages to say Google or Yahoo, or some receiving entity, there is no way to do that securely. So the miracle occurs phase, and I've presented this slide once before, really there is sort of this "F" score in the middle, there is no security.

So, if we go step by step for how a sending mail transport agent needs to send mail to the receiving server, it would first do a look up. So, you transmit your mail to your ISP and then your ISP takes it from there and has to figure out what machine on the internet it actually needs to be sent to. And so it would start by looking up, say, example.com's MX record, which is the DNS record that holds the mail server.

The second thing that it would do is that it would take one of those, so the example on the slide is ICANN.org's mail servers, and you'll note that they're all priority 10, but a lot of times they're actually a prioritized list, and we'll see other examples of that in future slides. So, the mail server would pick one of those, it would pick the lowest value normally if they weren't all equal, and then it will look up the address record for it.

So the second part of that is it's going to look up, in this case it's the IPV-6 address, and then it's going to connect to it. The

problem is that it has absolutely no security to it so now we're going to get into next how to do this securely using DANE. So, in the past, probably five or six years, before the secured DANE connected TLS connections were possible, the only potential was that people could just try to do a TLS connection and hope that it succeeds, and hope that you're talking to the right mail server. So, why is the previous insecure and just as I said a second ago, essentially a man in the middle could talk even in the middle and convince both sides that there actually is no security even available or that you're talking to the right place, when in fact you're not, and there is no way to prove that the other side both could do a TLS connection and that you were connecting to the right TLS endpoint even if you tried to do TLS.

So this is where DANE and SMTP came to the rescue, which is IETRFC 7672 and to indicate my bias level, I am one of the coauthors of that document, so I'm clearly biased toward this particular technology, but I will try and speak generically as the morning goes on. So, what does DANE and SMTP do to this whole problem? This is an example looking up IETF.org, ICANN.org doesn't actually implement DANE and SMTP, unfortunately. So, IETF actually does and so it does the same sort of approach. It's going to look up the MX record, where do I need to connect to, what server do I connect to when delivering mail to ietf.org?

It's going to then look up the address and in this case it gets back a quad a IPV6 address and the third step is it's now going to look up a TLSA record associated with that receiving mail server. So it's going to look up _25._TCP.mail.ietf.olrg, and it's going to look up a TLSA record. In this case that resulting TLSA record will be a fingerprint or a verification record some kind, it's not always a finger print. It shows you two things, it shows you, one, that they do support TLS, and two, this is what you should expect to connect to. And because it's secured DNSSEC from the top of the root down or from a trust anchor down, you know that you're getting to the right place and you know that you must do security, so you shouldn't accept a connection which isn't secure.

Scroll up just a touch, and so this is essentially what I just said, which is that there is sort of this aha moment, the DNS proves that you should do TLS and that it does exist, and that's the forward path. This is actually I think one of the most important aspects of DNSSEC which is that it proves whether a record exists or it doesn't. So if there is no TLSA record you know that there is no guaranteed secure path to get there, you can still fall back to opportunistic encryption, meaning that you can try and connect over TLS, you just can't be absolutely guaranteed you're getting to the right place.

So, most importantly, DANE and MSTP working together provide proof of existence, they provide proof that you're getting to the right correct TLSA end point, and they provide proof when security isn't available. But the down side is that it requires DNSSEC, and as we well know from the counts, counts, counts and other presentations, DNSSEC deployment isn't entirely ubiquitous and in particular some large domains have issues deploying DNSSEC.

So, this is where MTA-STS comes from, MTA is mail transport agent, that's the mail servers, and STS I'm actually blanking on the acronym expansion at the moment. SMTP is one of the S's and Transport is one of the S's and I'm forgetting the other one. Strict Transport Security, thank you very much. I got all three of those letters wrong. So what happens if you can't do DNSSEC, and there are some providers that challenges deploying DNSSEC because of rapidly rotating domains or they're unable to for other reasons. This is where MTA-STS comes in and this straight quoting from the RFC which is 8461, the primary motivation of MTA-STS is to provide a mechanism for domains to ensure transport security, even when deploying DNSSEC is undesirable or impractical.

So the goal again is don't require DNSSEC, can you still do mail transport security at least in some fashion, and we'll find out

that it's a little bit weaker than the DANE version, that's actually why this document actually can't overwrite the DANE policy, so DANE actually becomes the trumping factor in this case. So, if you can do DANE the sending mail server should do DANE only, and if it can't then it can fall back to this. So again, if we fall back to sort of the same type of thing, we look up the MX record just like we did before, we look up the address record where to connect it to, and then there is a new, I'm now falling back to Google because Google doesn't do DANE and MSTP but it does do the MTA-STS version.

So the third step in this case is looking up the MTA-STS record which is a reference, there is only really two things in it. It's a text based DNS record and it must start with _MTA-STS and then the prefix of the domain that you're sending to. And there is a version number in it which is STS version 1, and then there is an identifier, which if you look at it carefully, looks a whole lot like date. So, what this really says when a mail server is trying to send mail, it looks up the text record and it says, oh, they actually do MTA-STS, the next thing it has to do is it actually has to open an outgoing HTTPS connection to actually go get the policy. So unlike DANE and SMTP where all the information you need is stored in the DNS, in MTA-STS some of the information is on an HTTPS server and is verified through the x509 certificate authority based mechanism.

So, it's going to go fetch this particular URL and note that the MTA-STS keywords in that URL are all specified in the RFC so you basically have to put your website here, there is no way to put it somewhere else, you have to put it in that sort of the path, where Google.com is the domain being sent to and so you prepend MTA-STS and you add dot well known slash MTA-STS dot text. I love reading URL's, it's always so clean and clear. And so when you go look up that example, you end up with something like this, and this is the actual record from Google that I looked up on Thursday or Friday when I was putting these slides together and we're going to go over what each of these different records mean.

The first one, the version at the top which is STS Version 1 in case they do protocol modifications in the future. The second one is the mode, so the mode states how production you want to be when you're deploying this. There is enforcing testing and none. So enforce means that you should not connect to the server if it doesn't match these patterns, so we'll come to the MX patterns in a minute. Testing means go ahead send mail anyway, but if you discover an error just log it for me so that they can go negotiate and actually fetch the logs later. And one RFC I'm not going to talk about today is 8460 which talks about the logging mechanism, I'm not going to get into that. The logging

mechanism actually works for any sort of TLS connection, not just MTA-STS, it also works for DANE and SMTP.

And then there is none. So none means essentially I'm removing, I used to do MTA-STS but I'm going to remove it. So if somebody fetches a none, they should know that MTA-STS is no longer being used for this domain. So the MX field specifies basically the list of legitimate MX connections that you should be able to correspond with. It's an exact match or there is a wild card that matches one label and one label only. So, this is Google, note that Google has an MX record of aspmx.l.google.com, and there is also names that can be in front of it, indicated by the start.

Note that the star record does not match the first one, so the star record has to match a label and it doesn't mean N number of labels in front of it, it has to be only one label in front. And then finally the other field is the max age parameter which is 86400 and it specifies the lifetime of the priority after you fetch it, that you're suppose to cash it and leave it around on disc so you can remember it. So you store this up to that long so you don’t have go make these HTTPS connections every time you're sending mail to Google, you can see if you have a previous copy cached, as well as you can cache the text record for the lifetime

of the DNS TTL, they don't have to be equal, the DNS TTL could be far shorter.

The whole reason for putting a record in DNS note that it's a much shorter lived connection is the DNS is just sort of a reflection of you should go refetch it, because if the DNS text record changes, even though your cache for the HTTPS server has timed out, you should know that you need to go get a new copy regardless of the fact that the lifetime age hasn't been reached yet.

So, if you ever want to change or delete your MTA policy, there are a few things that you have to do and the important thing is that you publish the new HTTPS policy first. Because there is actually sort of a race condition built into the protocol. If you publish the text record first, somebody may actually try and connect to your HTTPS server and get the old policy before you update it. So you should always update the HTTPS policy first and then go update your text record in the DNS. If you're going to delete, you're no longer going to do MTA-STS maybe because you can now do DANE, you need to set the mode to none in order to start the removal process, and you have to keep these records around again for the max age that you had set previously. So, whatever your max age is set in your MTA-STS policy, you had better make sure that is viable for that lifetime.

So, I'm going to move now to a little bit of comparison between DANE/MSTP and MTA-STS. As I mentioned before, DANE/MSTP is technically more secure. If you read the security considerations in the MTA-STS document which is about 2 pages long because there is a lot of detail in it well worth reading, it talks about that and it specifically says that MTA-STS should never trump DANE because DANE doesn't have a leap of faith type approach like MTA-STS does. So this is a very long table, there's a lot of detail in this and I won't go into it entirely, but both RCs are listed. Interesting, we already talked a little bit about the testing policy within MTA-STS which says I am only deploying this for testing, please log failures and let me know, but otherwise I'm going to deliver anyway.

In DANE there is sort of partial deployment as the other way of testing, so you can have one mail server which is secured by DANE and then fall back to a second one which doesn't have a DANE record and allows the mail to come through anyway. Some differences between the two, DANE/MSTP and MTA-STS, downgrade resistance, it's basically impossible to downgrade DANE because it's all DNSSEC signed and you will know if ever you are getting records that say you're not supposed to connect over TOS or you're not supposed to connect to the right server, and that is not the case with MTA-STS, because the initial text record that you go get or the MX record list hasn’t been signed

by DNSSEC because again, the whole point of this is for people that can't do DNSSEC.

There is sort of this leap of faith. You have to accept the records that you're originally getting, then go connect to the HTTPS server, and hope that nobody has messed with the records in the interim. Once you have collected that policy, once you have fetched that MTA-STS policy over HTTPS, then you're supposed to remember it for a long time. So that max age property is trying to protect you for a period of time in the future, but of course eventually those can time out. The other noticeable difference is scalability. If you look at how DANE records protect MTAs, if you have a mail transport agent that is accepting mail for 10,000 different back end hosts, in order to protect one domain you add a TLSA record and then you add a TLSA certificate, and you're done. MTA-STS requires one more, you have to add a TLSA record, an HTTPS site, and a text record, so there's three things you have to add. If on the other hand you start adding more domains and you're pointing to the same MX server, you don't have to do anything because that DANE record was already attached to the MX record so you're already done. So that is it, any questions or comments? Flames?

ANDREW MCCONACHIE: Andrew Mcconachie, ICANN Staff. What's the acceptable list of trust anchors for the HTTPS connection? How do you know if you're setting up an MTA-STS receiver, how do you know that someone is going to accept your trust anchor?

WES HARDAKER: That's one of the problems that we deliberately put into the DANE docs, because there is no user validation, there is no user to say click yes to accept anyway, so it has to be hard coded. The RFC does not specify what 130 you're supposed to believe, it simply says it's most likely to be similar to the browser list. But as you know, not all browsers actually have the same list, so that is absolutely a problem, that you may not be able to actually securely retrieve the policy if you're publishing under a certificate authority tree authentication mechanism that the sending MTA doesn't accept.

ANDREW MCCONACHIE: So in practice does everyone just kind of trust the Mozilla list of something? Is there one that people just trust?

WES HARDAKER: Essentially the Mozilla browser has its own list, but every operating system typically has their own CA list and it's often up

to the operating system, so Apple has a different list than Microsoft and Microsoft has the ability to do some dynamic updates and fetch stuff if it fails.

UNKNOWN SPEAKER: That's all true, but we all use Lets Encrypt.

WES HARDAKER: Lets Encrypt isn't accepted everywhere yet, but it's getting close.

UNKNOWN SPEAKER: I actually implement the DANE stuff and I implement most of the MTA-STS stuff, and I can tell you in practice they're both a pain in the patoot.

WES HARDAKER: All security is, unfortunately.

UNKNOWN SPEAKER: I think the lack of soft failure in DANE is really, it's an issue. It's like I thought I got everything working and the way I discovered it wasn't working is that my mother in law could not write to my wife because she's a Comcast customer and they are pretty

much the only significant provider in the world that actually validates your DANE mail server certificate.

MICHAEL CASTOVAL: Michael Castoval. Speaking from an implementation and this more DANE in general than DANE/SMPT, but when you do a validation check with DANE you are trusting the recursive resolver to do your DNSSEC correctly. I would almost wonder if it's better practice for a DANE resolver to go to the root and check the entire chain by hand, because otherwise you cannot see the RSIG records and determine if the TLSA record is correct. I know there has been some effort to extend the TLS specification to staple this and I also know that's currently in development. So I'm curious on what current best practice is on how to handle this, especially if you're dependent on an ISP provided recursive resolver which may or may not do DNSSEC correctly.

WES HARDAKER: Two different answers and I'll give my personal bias first. I have always been a proponent of pushing DNSSEC validation and recursive resolver all the way into the application for security based decisions, because I think that's the only way to know. The second thing is that if you read the DANE MSTP

implementation for things like postfix and I think exim specifies the same requirements, the codes doesn't force you to, but the specifications says you must use a 127.001 validating resolver and so their notion is that if you're communicating over the loop back address to your recursive validating resolver, that is sufficiently secure. So, they say that you're not supposed to use your ISP DNS server as a separate one unless it's running on the same host.

MICHAEL CASTOVAL: Okay, and my second question, more of a security point with DANE is on the list of advantages and disadvantages. X509 revocation checks should still be taking place under DANE because certificates may be invalidated by a certificate authority for reasons that a user may not be aware.

WES HARDAKER: If you go read 7671, I think, which is the operational updates to DANE, it talks specifically about when you should do relocation checks and when you shouldn't. If your DANE record is 311 you're basically saying ignore the CA tree, and I'm actually tying it to directly to the end certificate so you actually don't do relocation checks because you should actually replace that TLSA record instead.

MICHAEL CASTOVAL: Well, I disagree with that logic, because if the key has been revoked by certificate authority because it's been compromised, then if it's 301 that means the public key and the private key have been compromised, anyone can recreate that certificate, have a matching TLSA, or decrypt in flight. You're basically ignoring an entire security mechanism of X509 relocation.

WES HARDAKER: So that's why other TLSA records exist, that say I actually want you to go check the CA side. The type 3 indicates that I'm ignoring the CA side entirely, so you can actually have an unsigned certificate as an anchor, so there is no CA side to check. And so the point is that the TLSA record and the DNSSEC record actually becomes the relocation check so you can yank that.

Any other questions? I'm ending exactly on time.

RUSS MUNDY: Thanks Wes, excellent. And it's now time for the coffee break. Back in 15 minutes please.

[END OF TRANSCRIPTION]