LOS ANGELES – Tech Day
Monday, October 13, 2014 – 10:30 to 17:00
ICANN – Los Angeles, USA

SANDOCHE BALAKRICHENAN: My name is Sandoche Balakrichenan, and I work for the French registry. Before starting this presentation, I would like to say that I was not involved in this work. This work was done by a colleague, Stephen [inaudible], and since he was not able to come over here, I had to present. So the idea of this presentation is to give us about different events that happen during the attack on dot FR name servers. And if possible, I will try to give you some responses, otherwise, the object is to have feedback from you.

And if you want, you can contact directly Stephen, if you have further questions. So in September 2014, we had an attack, and what we… It was one of the biggest DNS attacks on our servers that we have found. This attack actually doesn't impact the normal common man who is using dot FR, but it was seen by the monitoring tools. So, the attack had a domain name, which is on the right hand side, whilst always the same, and on the left label, it is was [inaudible] random, well it was not [inaudible] random, but level names, which were used for this attack.

So as you could see, the SLD and TLD and [dot, dot, dot] remains the same. So when we tried to look at who is having this domain name for WHOIS, this is what we found. For example, what you see, the address here is that it's from a province from China. But if you can see the country, it is from Switzerland. So when we try to Google it, what is that? dafa888, if you just see that, dafa888.com, it is a [inaudible] site from China.

So as I said, it didn't impact the common people who are using dot FR, but it was seen by our monitoring tools. So from this example, if you could see, [query?] the director of the attack, the source ID is very common, I think everybody here, you know it easily, it was from a Google public DNS. And this nation was… Here what we have is one of the name servers for dot FR. It is d dot [inaudible] FR.

So, there was a question, I had discussion with Matt here last week, who pointed out this, why was it not gassed by the DNS already? When at the first time, the query had an [inaudible] domain, and why did it always go without [inaudible] servers on dot FR? So, I'll explain it in the coming slides. Here, I'm not sure I understand it completely, because Stephen sent me the explanation, Saturday by mail, so I'll try to explain that and if I'm wrong, please correct me.

So, this, what we see here is the instances of the name servers for dot FR and [inaudible] servers we use, and others are the [any cast?] instances, so what we will show here is only the one that is used, it is called [inaudible]. During the attack, there were many [inaudible] used, and the biggest, being the Google public DNS, what we found out was that open DNS was not used, and most of the attack, it was [originated?] from [Europe?], and some it came from Indonesia. We don't know why.

We don't have any answer for that. So there were some initial patterns that we see that normally it had TCP packets, maybe it is because, when it send, there was a [truncation?], and that for it to be on that, it tried it in TCP. Also the [inaudible] maybe it is the behavior of the resolver that it modified from, IPv4 to IPv6. So this is a sequence of events.

What we see here, I'm showing, is just for [inaudible] there are other servers that are also impacted. At 2:52 in the morning, the attack started around 60 [inaudible] per second. So one of our [any cast?] nodes, which is [inaudible] and from DNS [inaudible], there was one to 7% package loss. At around 7:00 in the morning, Arjen Zonneveld from SIDN, he pinged Bortzmeyer via XMPP and informed of the informed of the attack, and AFNIC team was involved in solving this issue.

And around 7:59, Netnod also informs AFNIC officially. So, what's the action that we done? Our department, legal department, it tried to pull down the domain name. And the domain was blocked. Even after the domain was blocked, the traffic increased, and it was approximately 200 [Kr] per second. And this attack lasted around 14 hours, and at the peak it went up to one million [inaudible] per second.

So, now we will come to the question, why there is… For example, when we send, we have around random at the left end side and others, [www, dot, dot, dot] dafa888 dot wf, is always the same [inaudible], and why, when it called [inaudible] domain a the first time, the Google public DNS resolver didn't catch it, why did it always send it to the authoritative servers?

So even Stephen didn't have the right answer. At the beginning, we thought it was because of NSEC or NSEC 3, because dot wf is signed. I just forgot to mention that dot wf is one of the [www] of France, and we, AFNIC, we manage dot wf. So, Stephen discussed with policy, and policy seemed to provide the answer. What it said was that, it was a behavior, the DNS is always, that we have…

It was a bug, we cannot call it a bug, but it was an intentional bug. Because he said, that rfc 30134 was not clear, that's we see it in all of the DNS servers that we have. And the reason is that, for example, when we have an [inaudible]… When we have a domain which doesn't have, it just records by itself, but it does [inaudible], for example, if you see [inaudible] FR, it is an empty, [long term bug?] Okay.

So, when you send it a query to [inaudible] FR, the response what you get normally is a no error. But some cases, the authority name server, it responds with a [inaudible] domain. So in that case, the resolvers, what it tries to see is that even if NX domain is seen, it wants to see whether the [inaudible] is existing. So it tries again to ask the authoritative name servers, so it's a type of [inaudible] behavior of the DNS resolvers, to again and again query the authoritative name servers to have the response.

So I think correctly assume the point, if I'm wrong, please correct me. So, what the solution is that, there was a draft by policy, if I could see.

Yeah, it's a draft we see [inaudible] very simple. The draft is expired, but it explains how the resolver should behave. And Stephen says that it's new draft to many [inaudible], which is being sent to the DNS working group, may also [inaudible] the resolvers from doing this.

Now, what you see is that, the initial review of the attack, what happened for different name servers used by, hosted by AFNIC. So as you can see, at the peak, it went up to more than one million [inaudible] per second. And if you see from the DNS [norm?]. You have, around like a [inaudible], packets, the impact of the [sellers] also by SIDN. We had around 92% of packet loss. But there are other green signals

maybe, but we had packet loss, the green turns to other colors, and then a small only if about, more than 66% of packet loss.

So we don't see much information here. So how the attack was mitigated. So, AFNIC informed Google, and Google black listed the domain. The information regarding the attack was published by AFNIC around 5:00 in the evening. So there are some lessons learned. At the beginning, undelegating, what we tried was we tried to block the domain, but that was a bad idea because it increased the query rate.

And during the time of the attack, we didn't have two teams. One to combat the attack, and one to safe-guard the scene of the attack. So that we could have [four or six] after that, we didn't do that. So that was an issue, we think, we are [inaudible] for us. Even if they are, we're using number of monitoring tools, the first information that came to us from SIDN.

So what, first… It is important for us to have networking in the community, keeping its contacts always so that if there is any problem, we are informed. As we thank the colleagues who helped us and who informed us, and there were others, also, like Google and dot lv and CERT, which also informed us. Then what we are trying to do now is to clear the level three internal laboratory to recreate these type of attacks and see how we can solve it before it is, there is a live attack on us.

So the final point is that, even if we have seen a number of DoS attacks for the different people, any of the solutions, but it's not solved, there is more work to be done. And if you want to have more information, technical, you can mail to Stephen directly. And if you have any feedback now, we can provide.

UNIDENTIFIED SPEAKER: Thank you very much. I must say, some of you may know, since London, I'm running a small probe on one of our name servers, which only does a part of our DNS, and I find 1,376 [inaudible]… I'm looking just now what days, it's the same day. So I'm going to start looking at this too.

But I think it's quite interesting to find out that, the protocol or something, [inaudible], and nobody is perfect. Any questions? We've got two mics. I'll take the two questions. You were first and then you.

JONATHON: Thank you. Jonathon [inaudible], Microsoft. Thank you for the presentation, that was interesting. And it's a question for yourself and also for the room really. And it's something that also dawned on me, I think, in one of the talks yesterday, was talking about the same question of, this, how do we handle these random query attacks that are coming through public resolvers, and therefore targeting the authoritative?

And, you know, that was an interesting presentation as well. It seems to me that, in a DNSSEC world, the authoritative will respond with NSEC or NSEC 3 record, which the recursive… It gives the recursive the information to realize that hey, there is a whole range of the DNS space here that there are no authoritative records for.

And by caching that NSEC 3 record, it could use the, the recursive could then use that NSEC 3 record to realize that hey, this next query that comes in, hey I've got a NSEC 3 already, or NSEC for that already, I shouldn't be passing this to the authoritative server. I'm interested to know if anyone has looked at that approach for protecting authoritative

by having the recursive server do this? And also whether it's something that potentially could be an inducement for DNSSEC adoption and signing zones.

Effectively, the recursive servers were doing this, an authoritative server would have a strong incentive to have their zone signed, because by signing their zone, they would be protecting themselves from this kind of attack.

UNIDENTIFIED SPEAKER: I'd like to comment on that. He pointed out that there is a draft [inaudible] that propose one of the, this is one of the approaches. But it was quite controversial at the time, in DNSSEC extensions.

MIKE: Hi, Mike [inaudible] from IFC. Like the gentlemen who just spoke before me, I wanted to refer to you to the presentation given yesterday by Ralf Webber, which described another complication caused by these same patterns. They've been observed by quite a number of organizations that have reported them back to us and other groups, but they're widely believed to be extortion attempts against gaming and gambling sites in the Far East.

And I think a lot of the damage being done is collateral damage. But we've been working on some counter-measures. Evan yesterday offered access to some experimental code that we've been working on. If people are interested in that, they should talk to Evan or myself about that. It would have not helped in your particular case, but for recursive

who are having issues with these queries, we're working on something to mitigate that.

UNIDENTIFIED SPEAKER: Okay. Thank you very much. Our next presenter doesn't, I don't really, unless it's really very important. I don't really want to take too many questions, so the presenter himself didn't write the presentation itself, he's just sending in, I don't think it is going to be really very helpful.

The next presenter doesn't really need an introduction. Paul Mockapetris was so kind, and responded favorably when I asked him to give us a keynote presentation, and I said he could speak about a topic of his choice for the duration of his choice. Please go ahead.

PAUL MOCKAPETRIS: Thanks very much. Actually, I've got suggestions that I should talk about the history, which I find pretty boring, and most of you are all familiar with the history, it's hard to say anything that's new about it, or even interesting or amusing. But I decided to compromise and talk about how we disrupted things in the past, and how I would like to see them disrupted in the future.

And if there are any bomb throwers out here that want to work with me on one of my paper pet projects, which I'll talk about at the end, feel free to walk up to me after the talk, or whenever you like. So, the talk is about disruption in the DNS, how we did it and how we might do it again.

And my thesis is we need more. I really have liked a lot of the technical talks that are here, but there is altogether too much, I think, in the DNS world about, there is hardly anything we can change, and we can't think about going off in some wild new direction. I think we should try and do more of that, and this talk is sort of about that.

So, how we have disruption early on. My original marching orders from Jon Postel came, he came into my office one day and said, "There is about five proposals about how to fix the host dot text problem, and what we would like you to do is make some small little thing that will replace host dot text. Just a small structure, maybe like this tent, or maybe like…"

The thing at the bottom is actually a parking structure, but we just need to be able park more of these names, because the garage is getting a little bit small, so it could you please, you know, go do something different. Now, of course, I just ignored those five proposals and went on my own merry way, and that's how we got to the DNS.

And you might say, "Well, okay, how is it possible that Jon or all of these other people didn't say, 'Wait a second Paul. You're going off the deep end.'?" Well it's because they were off doing very important things, at least important things at the time, like trying to figure out how card images would be shipped over TCP, because you didn't want to send all of those blank columns, you know. How would you pick them in? And things like how rodding would scale and all of that.

So that's how we had some disruption there. Now one of the things that I tried to do early on, this is an image of the first, I think, documentation we had about it, that I managed to find this report the

other day. So the, you know, the time schedule early on was in 1983, which is not here because I can't go on record saying this.

I got a delegation from the open systems, the SOI people about how I should stop immediately because I was hurting the progress of the field. And they literally told me, quote, "We will bury you," from Nikita Khrushchev. Didn't work out so well. At any rate, we managed to get things from the specs in '83. Redundant roots were operational, and that, I think, got a lot of stuff going.

Dot com got created. One of the reasons dot com got created, by the way, was that there was a bunch of people, who were from the government, who said, "Oh no, this commercial stuff shouldn't be put in there." And I said, "Well, we should have some way to go do it." And so then finally they said, "Okay. We'll allow dot com because no one will use it."

[Laughter] "And it will eventually wither and die." True story. You might say, well I won't tell that one because there is other people who would be upset. Documentation started to appear and we had some parallel process in resolvers. And one of the great ways that the system managed to get off the ground, because getting off the ground is one of the key issues, is that the root servers didn't evolve nearly as fast as some of the Unix code, but they were stable, which allowed the Unix people to make more progress.

One of the early implementation ideas, and this is again, from that same write up about how I did it, was to say that the structure of the whole DNS system, the resolver here, the zone transfer mechanisms and so forth and so on, we should segment them into these separate chunks,

and then different people could work on different chunks, and we can share it. All we have to do is agree on a shared database format.

Never took off. Zero interest in trying to figure out how to do the shared database format. And it turns out that this first DNS server also had the property that, you didn't have to reparse all of the zone files to restart the server, because they just kept blocks in memory and cleared them and did all of that kind of stuff in the paging system, which was an idea that came back about 20 years later. But it's interesting to see how some of the stuff got going.

So the whole idea here is that the RFCs 882 and 883 would be the basement, if you will, and then we would build this beautiful organized cathedral above it. Some of you may recognize that as a work that I think was started about the same time as the DNS, at least the most recent construction of Gaudi's Cathedral.

It is still being built today. So, again, a little DNA from the initial stuff. What was unique, I think, about it at the time was that the UDP and the server redundancy principles were at least an unique recipe. There was very few really unique ideas here. And I would like to thank [ARPA] for supporting ISI and Berkeley and some other places, to actually get this innovation going.

So that was going along. And in 1989, sent a proposal into the National Science Foundation, to say, "Well let's do some stuff to innovate in the DNS area." And things I proposed were to fix bind, and then address things… Add features like incremental updates, security, and also the idea that we would index the Internet of the time, and build in the DNS public indexes so that you could find things.

And then also deal with abuse. It turns out that some microsystems had a release of their code which created something called the main name screaming, which was sort of accidental DDOS, and so we were going… We said, "Well, we're going to have to address the DDOS question."

So we sent this all in. This was again in 1989. So NSF reviewers came back and they said, "Excellent. Very good, very good." Okay, and if you have ever been on a NSF review panel, this is sort of the way that you give somebody the kiss of death. Because there must be somebody that got excellent, excellent so you know, this work wasn't seen as particularly important, and the idea of indexing the whole web, in particular, was rejected by one of the reviewers as being something that was totally bogus, because it was too big.

This is the point which I guess I should have, you know, walked over to Klein or Perkins and see if they would have sponsored a commercial effort, but you know, the wisdom of the academic community was that, all of this other stuff just wasn't important. All right. But excellent very good, very good actually has one other part to it, which is, and I still have the reviews, and I think I know, I know where two of them came from.

At any rate, and they don't know I know, but they will after this talk. At any rate, the NSF result was, they couldn't decide. Well, so much for planned evolution. So, you know, we entered the era, I think it's wonderful. This is the family tree of DNS RFCs. They each have… So, you know. People always say, "How did you invent this whole DNS?" And I said, "See those like three boxes in there? Those are mine, and the rest is stuff that you guys out there have invented."

Which has its… I mean, it's frankly phenomenal about this. I occasionally pulled… Bob [inaudible] and Vint Cerf's chain, because you know, they were the fathers of the Internet. And that… And then I went to Google and they actually there is more PhD thesis revolving around the domain name system then transmission control protocol, or Internet protocol.

So another big success of the DNS seems to have been to create PhD thesis. All right. How about the future? Enough about the history, and I hope that I amused you at least a little bit, maybe told you something that you didn't know about the history. I'm going to talk a little bit about three ways to disrupt, and I'm not going to talk about any of the usual suspects, about like, should we just send DNS queries over TCP and get on with it?

Or those kinds of things. Those are very important work, don't get me wrong, but it just doesn't seem disruptive enough. So let's, first of all, let's think a little bit about other inventions. And here is four inventions, or landmark efforts. And the one that I respect the most out of these four, I'll let you think about it for a minute, you can figure out which one you respect the most, it's the wheelie bag.

It turns out that if you look back in history, form the time that the first Zeppelin carried commercial passengers to jet aircraft was about 40 years. So about 40 years of evolution to get to the airline industry as we know it. It took another 40 years for the inventor to come up with the wheelie bag. All right? So I believe that there is probably innovations out there that are simple things, like the wheelie bag, that we can, if we put our heads to it, we can do.

I have nothing against innovations like the Concord, except well, gee, there aren't any more anymore. And there are other great inventions in other fields. It's not all about the Internet. The Polio vaccine was very important, but I think one of the things to do was to try to think about ideas that are a little bit simple.

And I have the greatest respect for all of those people who were doing all of that math yesterday, but, you know, I think there are some simple things out there. Let's take a look at the DNS basic algorithms. The initial design was purposely minimal, actually scraped away everything that it didn't seem to be absolutely essential.

So the whole idea, for example, about how you find stuff in the DNS, is you go to the top and then go down. Go to root server, then you go down. That was the only two-step process I could think of. All of the processes, other than that, involved three steps or more.

So why don't we think about changing that? You know? Maybe you should just be going up from your local domain, and then go down. One of the things about a tree is you can take any node in the tree, hold it up, shake it, and make it the root because the rest of the tree falls down underneath. You know, maybe we should just think about new ways to transit it.

And that involves different trust anchors and the whole idea about having a single trust anchor for the whole DNS seems to me to be fundamentally broken, as long as you can't identify a single thing that everybody trusts. So, you know, we should, I think, relook at some of those things.

There is a recent set of efforts to do query minimization. And it went out for comment on the DNS working group list, and lots of people had said plus one, and I couldn't decide what to do, because I was very tempted to say plus one, what's there is wonderful, but I also sort of felt that I would have to include in it minus one. Because the whole idea about getting rid of [inaudible] to caching, without guidance about where it might be effective, or for what data it might be effective, and so forth, just kind of bothers me a little bit.

You know, I believe in privacy of communications, and I think if I had to make a list of people, I would love to just have a chat with someday, Snowden's name would be on the list. But I don't think we, opportunistic caching is a very valuable thing. I think we need to figure out where we need to preserve it, and at least say that before we say, "Let's minimize everything."

My favorite example of something that I wish some of those RFCs had addressed is, you know, the fact that a query has account of questions, was because I thought people were going to put modifiers in there to guide the search. Never happened. I think we ought to do that. If you take a look at a lot of what's going on in information centric networking, they have both names and modifiers and say, what page of the data? What frame of the movie? Whatever that you want.

I think we should think about ways to add those capabilities. And lastly, I wish Steve Jobs was around because he famously said, "Get rid of floppy discs. Get rid of this legacy technology." We need to figure out how to kill some of this old, backward compatibility that is just holding

us back. About a week ago, I went to an information centric network's conference in Paris. And this is fascinating stuff.

And I dearly love it. In some ways, it's better than the DNS. I was wondering, one of my theories was, could it be used as a replacement? But they have their own set of issues. They want to go and replace the infrastructure, so it is all of these people who are building these routers, but they have to replace all of the routers to have it work, and it's going to be hard.

And the other thing that they do is, they believe that they have invented, well you should look up stuff by name, rather than by address. And I'm kind of going, guys, we're here. We can look up stuff by names. And they believe that you should take the whole naming structure, and then plug it into OSPF. And that, you know, you'll be able to route it.

But if you have billions and billions of names, you know, you need a sub-linear, less than an order, and an order to be able to do this router. I mean, how is it going to work? So I think one of the ways we might do is to think about how DNS might serve as the routing basis for these information centric networks. Is there a way to blend it in and move forward to a name based, a totally name based future?

They have this, again, better technology for segmenting the data and delivering it, but it also has a bunch of stuff about the paths have to be symmetric and so forth. It's really a bunch of genius, and then a bunch of, well, really odd restrictions that I think decrease from its practicality. ICANN had me do a strategic panel and come up with some recommendations, and a lot of these weren't new.

Like one of these things is we need better ways to distribute the root, and there has been others, at least a couple of proposals out there now about various ways to go do that. I think that's great. I said, "Well, gee, why don't we just figure out how to replace management of the root with some set of distributed algorithms." Which, I think, the technology used in bitcoin and name coin and so forth, could be used to manage zones.

It doesn't have to be a physical locus, from which the formation of the root zone or whatever, happens. You will have to decide on a shared distributed algorithm and what the rules should be. So I really look forward to being able to go to the politicians some day and say, "Okay, you wine about how ICANN is in LA, that's terrible. But okay, we're not going to move it to Switzerland, we're not going to move it China, we're not going to move it to all of these jurisdictions, we'll just make the jurisdiction banish. Okay? But, you have to tell us what the rules are you want. Okay?"

So that, you know, all I hear from a lot of people is, "You've got to move it out of LA." Okay, but I want to… If you tell me where, let's move it to nowhere and lets do distributed management. So this is my current project where I'm trying to disrupt the world. And I call it algorithmic contracts. This is my personal favorite.

And so the idea is to implement zone management using rules and logs and so forth. No jurisdictional locus. And you have one or more people that will just go look at everybody's, you know, request logs and figure out what the current thing should be. The fly in the ointment that I

haven't figured out how to deal with is, if you want to have just one signature for it. Well, how do you do that?

And there are some [inaudible] signature stuff, and you know, the mathematicians in the audience may explain that to me after, you know, during lunch or something, I'd love to hear. But, you know, this is one of the things that I'm working on. And furthermore, extend this to other applications, number portability, for example, the database, contact sharing and so forth.

Figuring out how to organize aggregates of DNS data, which is really what we're talking about when we put together the root, and extend it to other applications. Okay, so now you're convinced that I'm crazy, but that's okay, I got a lot of that over the years. But, if you were interested in helping me with this disruptive idea, fine. If I've inspired you to a different one, that would be great as well.

So thanks very much, and I hope I'm on schedule.

UNIDENTIFIED SPEAKER: You can give him a hand. [Applause]

UNIDENTIFIED SPEAKER: Actually, we are ahead of schedule, so any questions will be taken, and if we have some time we can take the questions from the previous presentation anyway.

Everybody is speechless? All right. Thank you very much. The next presentation is Casey Deccio. I have seen him in the room, there he is.

He will speak about DNSViz. I think most of us have seen this. You plug in some domain name in and gives you a graphical analysis.

It uses very old tools. But uses some tools called Graph Whiz, I think? Which allows you, in a programming way, to describe relationships and then it does the graphics. And I was quite interested in this because I use a similar software to determine exchanges in registrants from one registrant to another, when we have an issue about this.

So I found this topic quite interesting, and when he set up, and I'm just making small talk so he can set up, without further ado. We are well in time, so if questions, if there are questions, we will take questions.

CASEY DECCIO: All right. So I'm going to talk a little bit about DNS… I'm going to talk a little bit about DNS analysis and visualization, and some of the history behind what is currently the DNSViz site, some enhancements, and then some downloadable and installable tools. I'm also hoping to invoke a discussion, or at least some questions regarding feedback and some direction from the community that would be helpful in terms of those that have used this, or are interested in using this tool and functionality.

So, just by way of history, and some of you may be familiar with this, and some of you may not. But DNSViz has been around for several years. It began in 2010 to kind of fill the gap with understanding this system, this tent system that Paul described, that has, had some complexity added to it in trying to understand the different components of the DNS, particularly for a different installation or deployment of a DNS domain.

Okay? So it was launched by Sandia National Laboratories. And then in 2011, we began archiving some of the deployments as we were monitoring them, so people could go back retrospectively and look at what things look like in the past, to either fix things or see changes along the way. So it allowed kind of this timeline DNS, I want to say an equivalent, but trying to be similar, at least for things that it was monitoring along something like the Way Back Machine, right?

So you could go back and see things over time. Okay, there was a major database backend rewrite in 2013. And then in 2014, this project was adopted by Verisign Labs, where I am currently employed, so we, it could continue to be worked on and improved, for the users that use it. So as part of that, the analysis engine has been completely rewritten, almost from scratch, and it's… So one of the things, when this was, when we originally wrote DNSViz, so you can see over on the right, kind of the graphical output, well in generating the graph itself is where a lot of the analysis was done in terms of understanding where things were broken, were things needed to be, where there were other problems, and kind of giving hints to the users to where they needed to be fixed.

Unfortunately, that made it difficult then if you wanted to have some sort of, for example, restful API, or some other programmatic way of approaching it, rather than just the graph. And so, and also it made it, the way it currently was, it was basically tied into the database back ends, so it was not suitable for a local installation for just running it. You know, if it were to be, for example, downloadable tool that you used.

So, those were some of the major improvements that were made in terms of this analysis rewrite. And then, in addition to that, the hosting has now changed where it's being hosted now, it's currently a Verisign Labs, and there is a release underway. A zero dot one dot O release, which means that I am interested in feedback to kind of gear up to the next release cycle, but this is currently where it is out now.

So, the next couple of slides are going to refer to what is currently the tool functionality, in other words, what an user would download and install and use, okay? It primarily consists of three different tools. The first of them, being a command lite tool that I call DNS Get. Okay, the idea is that this is the part that does the online analysis, meaning you're running it from your machine, you're going out to all the different servers necessary to perform this analysis, and asking those series of queries with different parameters.

And by default, it will do things like, you know, try with a high UDP max payload size, also try with a low so you can actually see the results from either side. It will do TCP, it will look at the SOA for zones. If it's a name that returns a see name records. So in other words, it works not just for zones, but also for individual domain names. But whatever it takes to go through the analysis of this name that it is given, it will go through all of the servers necessary to do this analysis, one or more names actually.

And then the output of that, for after querying all of these servers, is this Jason blog essentially. And right now it's a file output, you so output it to a file, and then you can do something else with that. Which brings me to the next command. Yes.

UNIDENTIFIED SPEAKER: I notice you're getting a bit restless. The network seems to be down. So you notice.

CASEY DECCIO: I thought that was because everyone wanted to listen to my talk. [Laughter]

Yeah, it turns out you actually do have to be online to perform the online analysis, I didn't mention that part. [Laughter] But it's okay. None of you have to do it right now, you can just, you'll have to wait for that magic moment.

Okay. So, once you've performed the offline analysis, doing all the queries and getting all the responses that you need back, the next command that you can use is DNSViz, which, as you could imagine, takes all that offline analysis and creates a graphical representation. Okay? So you've taken that Jason blog, essentially, and you input it and then you get this output. Okay.

I borrowed some of the command line conventions from various other command line tools, in this case dot, which I don't show all of the options here, but to determine which type of output. So if you want HTML, or whether you want PNG, you could do it a couple of different ways. You could do HTML that it will include J Query for Java Scripts, you can have this kind of interactive mouse over type interface. Okay.

And then the third one, which is, this is a little bit, well this is new. The graphic was available through the web interface that you can currently and get at DNSViz dot net. What is not currently available is this online analysis that basically gives you a Jason output of status, and when I say

status I mean, okay, you've done the offline analysis, or excuse me, the online analysis, pulling in all of these queries, now this is kind of like the structural version of the graphical analysis that you got with DNSViz.

Okay, so this is more machine parse if you wanted to go in and actually parse it, now you've got a Jason representation of this status. Okay. So you go in and do all the analysis of the things like signature, validation, your NSEC or NSEC 3 validation. And DS and DNS key validation. Looking for things such as server response, and so forth.

All of that would be in some type of a serialized format. Okay. So, just, I've already talked about a couple of these features, but let me go down the line of some things that I've enumerated in terms of things that it can do for you. So it will automatically detect IP 6 connectivity on your own machine, and then if you have it, it will go ahead and do IPv6 connectivity tests on the remote servers as well.

I already mentioned that it works on arbitrary names, not just on zones. It follows [inaudible] dependencies, and then I also mentioned negative responses. Some of this is new in terms of doing negative responses and following authenticated denial of existence, looking at an X domain versus no data versus NSEC, NSEC 3. And it's actually come up helpful in the last couple of months with some issues that have come up in troubleshooting.

And not only that, there is, in terms of these three items, it doesn't show it here, but in the, both in the visual, the graphical, as well as the structural analysis, it will break down, for example, looking at the NSEC chains so you can see kind of the output and why things don't match if

something is broken. Like, you know, if the visual, you just don't get a redline, it actually shows you why things are working.

And so hopefully that is helpful. And then, supporting D name and wildcard, isn't exactly new but it also shows those structurally and in the Jason output, in terms of whether they're correct, how they apply to the given name, and follow those as well, D Name follows to the name that it points to.

Okay. Some other features, and these are mostly related to the command line tools themselves, as opposed to the output, but doing a stub versus full analysis, I call the stub meaning, just look at the name and it's direct parent ticketed delegation, as opposed to doing the full chain, because often times you don't really care about the whole thing unless you want a pretty graph to show your manager or something, that it works.

But you really just care about your delegation down. That is, your records and the parent, the DS records that are in the parent, and the NS records that are in the parent, glue records and so forth, and then from there down.

But, you can… So default is just what I call the stub analysis, but optionally you can go entirely up to the root. Okay, explicit delegation, if something isn't delegated, for example, already you can put in your own servers there, so you can test it out before it has been delegated, or if it's a test delegation, or whatever.

And somebody actually asked me a while ago, "Can you do this alternate root?" Sure. With this option, you can do that. So if you're

interested in doing that. Okay. DLD, if, it turns out there is still quite a presence of DLD users, and I say DLD, I'm referring to the ISC DLD, so if you want to, you can actually specify that and it will query that as well.

And then there is some support for parallel analysis to make things faster. So if you give a list of names, but I admit that it is not optimized in terms of the way it actually works, so use at your own risk. It consumes a lot of memory. Okay.

So this is written in Python, and this is kind of the way to get to things, so there is… This is the zero dot one dot O release, there are three dependencies. So there is DNS Python, pie graph is, and [inaudible] crypto. And I mentioned, in the source, that there is actually patches for supporting DSA, ECDSA, and Ghost because they're not, with the current functionality of [inaudible] crypto, it doesn't support it.

I have a link here to the, a temporary pre-release version of this, and the reason why that is, is because we're working out some final logistics that need to happen, anyway, just some logistical items. But for the purposes of this, I have put a temporary location to get a prerelease of the code, and I am very interested in those that are, that would be willing to pull it down after the network resumes working.

Oh good. Great. So that you can take a look at it, try it out, and I'm interested in whatever feedback you have. Okay. And then it installs like a regular Python program, set up build, set up install. Okay. You can install however you would like locally or globally, and then the license is GPL2.

Just to give you an idea of some of the output, so this is the top item here. So this is what it would look like, and the command line, of course, at the top here you would see what it would look like when you do the online analysis by running DNS get. Okay, first, and then you output it to a Jason file, and then DNS [grok?], I have this dash L error, which basically means, filter out anything that is not error and above, and so in this case, I get nothing wrong with VeriSign labs dot com, coming from the Jason file that I previously retrieved using DNS get.

And then if I do the dash L info, I can get a little bit more information, such as the status of the name, which of course, in this case is YX domain. It will dive down into all of the queries that it made and look at the R06 status. Go on. It will look, it will try things like trying to add a random name that doesn't exist, and therefore get the, make sure the NSEC 3 records work, it will return properly.

Okay, it will of course, look at the R06 on the NSEC 3's as well. It will look at the keys themselves, looking for any errors in the keys, and this is… Again, some of this is still a work in progress. There is more to be done here, but this is, it looks for certain things right now, and will look for more in the future. Things that may bite you when you get your zones out there. And then, of course, looking at DS status and delegation status in general. If there is something broken between your parent child, whether it's relating to DS or relating to setting authoritative answer bit, some resolvers handle that differently than others.

And there has been some behavior changes as well, so at least it will throw something up saying, "Hey, there is a problem here." There is

also a debug mode, which is the default, in which you get a lot of output. Okay, so that's what it looks like. Okay. That was a quick view, what the functionality looks like in terms of just running DNS get, followed by DNS [grok], and of course, DNSViz would output the PNG or whatever.

Okay. So let's see, there is some, when you install this, it actually installs a DNSViz module, a Python module that becomes a library if you will, that you can use. So that the scripts that come along with it are, implement some of the functionality of those. Now there is no well-defined API or anything at this point, but just so you know, there are some lower level things that are of use.

Okay? So there is, for example, sub-classable DNS query that includes using these query response handlers, so you can do things, you can basically define your own DNS query. Meaning, for example, a recursive verses a non-recursive, DNSSEC versus non-DNSSEC. Think of all of the options and bind, and then you can build a sub-class of DNS query fairly quickly, and then deploy that for your own code or whatever type of monitoring you're doing.

So that's the kind of thing I was looking for in this, and that I built, because I didn't find it elsewhere, a quick way of being able to do that. Okay, I've already mentioned this. I have, I am soliciting your feedback and help, in terms of usability, you know, how things look in terms of the schema. I'm open to suggestions and community input to help get a better tool for making the DNS more stable in your individual deployments, and to reduce overall errors, and to increase availability.

Okay. And error codes and descriptions, the level of [inaudible] that I've talked about, okay, feature requests, bug reports, all welcome. Just a note on future work, so adding in some regression tests or improving the ones that I currently have. The web front end is actually using the new engine that has been rewritten, the current, if you go to DNSViz dot net, it's using the new engine, but it still looks the same as the old one.

An improvement will be made there to migrate away to a new look and feel that matches the tools functionality. Okay. Let's see. A recursive implementation, the new analysis rewrite has facilitated the tweaks of the code that would make it suitable for not only querying the authoritative servers, but also recursive servers, which I think is almost important because, that's where you're getting…

If you're having problems, that's where you're having your problems, is the recursive. And maybe because of a problem with the authoritative, but you don't know exactly what is in the recursive at that time, and so that will be, that has been facilitated and will be implemented, hopefully soon. Okay, arbitrary record types is something yet to come, and so you can plug in whatever record types you want, not just the ones that it detects it should make.

And there is also a groups mailing list, which currently has no subscriptions because today is the first day I told you about it. But, I'm interested in people, if you want to further this conversation and have some feedback, you can either send it to me directly, or bring it here to this list here.

With that, I am happy to take questions, if we have time.

UNIDENTIFIED SPEAKER: Thank you very much. You can also give him a hand.

[Applause]

Any questions? [Jay Daily].

[JAY DAILY]: No question, but just thank you very much for making this open source and letting us all have a chance to use it. Thank you.

CASEY DECCIO: Thank you, you're welcome.

UNIDENTIFIED SPEAKER: I don't know, no this is not company policy. I work with Jay. Thank you very much. Whatever you need to contact me for, asking for the call, the query, make it available, excellent. And I have some happy people working this [inaudible], saying, yeah, yeah, finally.

The thing is, how you are going to handle the feature request and possibly [bunches?] stuff like that?

CASEY DECCIO: Yeah, it's a great question. So right now, the code is hosted on a Google code repo. And it's probably going to be migrated to get hub, so I've refrained from doing any type of ticketing system there, and I probably will wait until it has migrated to the new place, and have an official URL. So right now, you're noticing the slide I just put down in the current,

you know, URL for the download of the, you know, tar ball as opposed to the actual source.

That will be there at some point, but I just, it's not ready yet. So I apologize for that. But anyway, that's the way I plan to handle this, is through something like Get Hub. So that's the plan.

UNIDENTIFIED SPEAKER: All right. Thank you very much. Next will be Francisco Cifuentes. [Applause]

FRANCISCO CIFUENTES: Just bear with us for a second.

Hello. My name is Francisco Cifuentes. I'm a research assistant at NIC Chile Research Labs. And I'm going to talk about DNSSEC in particular, about a new management tool. Okay, so.

To solve security complaints of DNS, it was proposed the security exchange, as you probably know, DNSSEC, and to satisfy its needs, a DNS operator has been using hardware security modules HSM. It is a specialized hardware that manages cryptographic operations, and has special security needs.

For example, some level of the FIPS, one for each standard.

But they have problems. HSM are expensive. The cost goes from 50 to $50,000. But if you want to have a good security level, you will have to pay for it. For these reasons, small institutions may want to deploy DNSSEC, they cannot buy them. So what if we could achieve a good

security level without paying that much, or what if we use old and not in use hardware, and we achieve a good security level not paying at all?

So here is where we come. We propose threshold cryptography based HSM software, software implementation of a HSM, based on the open DNSSEC function, the open DNSSEC, needs, I'm sorry. So, first I'm going to talk about threshold cryptography. I'm going to do a little [inaudible].

So, in this scheme we have two different types of [inaudible], the nodes and the [signature dealer]. In this scheme, the [signature dealer] distributes the cryptographic work to the nodes, and it's a [machine] who joins the nodes at work. In this scheme also, we have a RSA key, split into a set of [inaudible].

RSA [inaudible]. And each node, it starts the key in a private way, and it's [only] authorized now that, [too easy]. So, when the [dealers], they receive a signed requests, it forwards the request to each node. Then the nodes virtually signs the document, and send the result back to the dealer. Finally, the dealer joins him, and if you have end nodes, only end nodes plus one verified answers are needed to generate the document signature.

To generate the signatures, the signature [dealer] generates a full RSA key, and then it is plated and sent back to the send, sorry. And the signature [dealers] sign it to send to the keys to the nodes. I'm sorry, I'm sorry.

The signature [dealer] send the key [signatures] to the nodes. And there are stored in a private way. Okay. This scheme has special security properties. It is secure, fault tolerant, and robust. It is secure

because a subset of the nodes, I'm sorry. It's secure because no one holds the complete private key, and that gives the security properties.

And it is fault tolerant because a subset of the nodes don't fail, and the whole system will be running in an okay fashion. So, and it is robust because failures on [inaudible] reviews, implementing the nodes in different programming languages, and running the nodes over different operating systems. So. Now I'm going to talk about the HSM use in open DNSSEC.

So, this is the HSM typical usage. The HSM typical usage is having an implication that uses the benders PKCS 11 library to provide, to privately communicate to the HSM. The PKCS 11 API is an API that the application uses, I'm sorry. Is a standard that the application may use to communicate with the [inaudible] security model.

And this is our solution. In it, we use [threshold] cryptography in order their security needs, instead of physical tampering. It works as a distributed system with different things in different places working together.

And here we have the open DNSSEC Architecture. In it, we can see the HSM that it uses to sign the DNS registry. And this is how we use our solution in this scheme. We can think it up as a cloud that uses the end work to communicate with open DNSSEC, but if you will look further in our solution, this is what we see.

A distribute system that implements the PKCS 11 API in order to make open DNSSEC, use it in a transparent fashion. The PKCS 11 library sends a [inaudible], the dealer has to execute through Zero MQ sockets. Then

the dealer creates a new request and send it to the nodes through a new Zero MQ socket. After the nodes have finished the work, they send the request back to the dealer, and it joins the whole nodes at work.

Now I'm going to show you an experiment that we have done, and it's [spreading] in our results. We have two configurations. Typical computer, and another in another configuration we use [inaudible]. We cannot [a matchings] of the same types, of the same type through our DWLAN. We [lag] lower than one second.

This is our raspberry PI cluster. It's our, right. And, in the experiment, eight nodes try to sign the zone registry. We measured the average time of the generation on the original 1,000 are our signatures, are our site signatures. In order to compare our results, we measured the same experiment, but you see stuff they just send.

That is also a software HSM solution that was made by the open DNSSEC developers.

So, here we have the results. Normally in a typical desktop, our solution to one [more magnitude order] Soft HSM to finish the work. These are raspberry PI, our solution between two, one to two more magnitude orders to finish the work. Maybe the results do not look very promising, but in the context of DNSSEC signatures, it doesn't have PI overhead, and it does, it is a fraction of the HSM, of the same security level.

As you see, our solution may just zero dollars, that if we use old computers that were not in use, using raspberry PIs, it would cost only $280, and it's a fraction of an HSM with the same security level.

Okay, so, but there are some implementation problems. We didn't know how to do [inaudible] in the [inaudible]. It's needed because the signature [dealer] can retain memory, the hosts [teacher said].

So, we this, we have future work here. First, we have implementation diversity. That means that implementing the nodes in different languages. We have managed this by doing the node implementation in [inaudible] and Python programming languages, with that can run in several operating systems. But, it would be nice to have other implementations.

And our future work is developing the full distributed threshold RSA, cryptographic scheme. And with this, we could solve the memory server [inaudible] problem, because the RSA [teacher] set will be generated in the nodes themselves.

Or we can use GPU to accelerate operations and doing replication or migration in order to mitigate one point of failure that the signature [dealer] needs. With the threshold, with fully distribute threshold RSA, it can be used that as a scheme that's replication or [inaudible].

So, with this, I finish the presentation. My name is Francisco Cifuentes, and you can talk me thour NIC Chile. Thanks.

[APPLAUSE]

UNIDENTIFIED SPEAKER: Thank you very much. Can you turn the mic off while we're not using it? Any questions. Okay, let's start with Rick [inaudible], because he's closest to the microphone, and then Sebastien.

RICK: First of all, thank you very much for doing this and sharing this with everyone. This is really interesting work. My question is, so do you intend of trying to… Or do you have interests from some to try to productize this into something has the [phip?] certification? That's a process, right? As you know.

FRANCISCO CIFUENTES: It is very difficult to say because we want to do this, and we want to give this to the community, and we give this as an open source product.

RICK: Yeah, but that's why I was wondering, has anyone approached you? Because a lot of times, I run into people that just want a solution, right? So they're willing to pay. They would just like to be able to just buy something.

No one has approached you? Anyway, wonderful work. Thank you very much.

UNIDENTIFIED SPEAKER: Hi Francisco. So you did your testing, you have all of your raspberry PIs in the same place, which actually defeats the purpose of your

distributed cryptography. So have you tested how the [inaudible] of the system is effected by delay, by network delay, or network loss?

FRANCISCO CIFUENTES: Yes, we have tested this through Amazon [inaudible] in two instances. There is no difference, only because the nodes communicate, I'm sorry. There is no difference because the main work is done in the nodes themselves, and that is the, where it takes longer. So, the delay doesn't effect, because the communication doesn't… I don't know.

UNIDENTIFIED SPEAKER: I can follow, obviously one [later], but if you have the coordinator, and you have the nodes, and I'm thinking, for example, an architecture where you have a set of nodes in one place, and build a set of nodes in the other place, they need to go and do their work internally, and talk to the coordinator.

But what if the coordinator says, again, in a different place, 100 milliseconds away from this? So I'm thinking, what is the effect of network latency on the throughput signatures?

It effects. Because, for example, one of the things that we're talking is, so the AEB keeper, one of the very expensive, [inaudible] you can have more than one and you can use them with a load balancer. So the architects of the AP has, they have a load balancer. And you have one of them, like 10 or 12 milliseconds away from the other, the platform of this whole [inaudible] is unusable.

FRANCISCO CIFUENTES: No, no, no it's okay. The [inaudible], it's reused if there is delay, but if we made this replication or immigration, we can mitigate that too, because the nodes, the work can be done in the nodes where there are more in a charted distance to the work.

UNIDENTIFIED SPEAKER: Okay, yeah. We'll discuss this later. Thank you.

[APPLAUSE]

UNIDENTIFIED SPEAKER: Next will be Aziz Mohaisen, who will speak about .ONION leakage, which I find a really cool topic.

We're presenting from the remote, so sometimes the picture is not as good as it could be, but that means the remote viewers can see the same picture that we're seeing at the same time.

AZIZ MOHAISEN: So hi. I'm Aziz Mohaisen I'm a research scientist at Verisign Labs. And today I'm going to talk about a measurement study that we have done for the state of .ONION and other roots, and this is joint work with my colleague Matt Thomas, also from Verisign.

So, the talk is going to be as follows. First we will talk a little bit about the global DNS and private namespace usage. Then general study of the .ONION from the root servers and A and J, with some longitudinal analysis that include understanding the nature of this handle that we

had used, understanding the volume and diversity of hidden services, classes that they belong to, most requested services.

Some GEO mapping and some ASN network level mapping. And then some even correlation. We will compliment this with some trends from the in the life of the Internet [inaudible]. Some investigation into the possible root causes, and the risks that are there and then concluding remarks. Questions and answers, if you have any.

So perhaps this is the wrong crowd for having a slide, but I'm going to just try anyway. So as all of you know, the global DNS is a hierarchical system, with a number of root servers, groups of root servers, 13 roots numbered or A through M. And they are responsible for answering queries for TLDs such as com and net, among others.

However, many of the installed systems to utilize, non-delegate top level domain names for their own internal namespaces, such as corp home and many others.

And the queries, often time, well always, queries to the root for those TLDs result in ND domain responses. On the other hand, the delegation of new gTLDs has spurred some more critical studies NXD related [inaudible] NXDs at the root, highlighting some name collision risks, as has been started with this workshop organized by Verisign.

One of the studies that we look at today is the leakage of .ONION to the root. And if you're not familiar with .ONION, the next slide is go over like where .ONION is being used. So it's used in this service by TOR, its hidden services. So TOR provides [inaudible] for both users and

services. For services, there is a server [Bob] who wants to not only serve [Alice] with content, but also wants to disguise his ID from [Alice].

So what [Bob] would do, he would create an X number of introduction points, three of them, and then the user himself at this [inaudible] server. And the user himself under a fancy name, that is a hash with certain contents, with extension [inaudible], so it is a lock out to the system.

Then Alice supposes that he is interested in learning more about Bob service, she would query a little bit asking for the, more information about Bob's service, then the [inaudible] would respond with introduction points. Meanwhile, Alice would create rendezvous point that would be like the proxy for communication with Bob.

And then using the introduction point, Alice would send back to Bob saying that, well here is the rendezvous point. Bob will send back some [life nodes?] to authenticate himself to Alice that he is the real Bob. And then communication goes as regular through the rendezvous points. All of those slides are actually just from [inaudible]. The idea to take away is that only, like .ONION is being used for addressing and naming, and the TOR work, that's supposed to be in the [inaudible].

Not supposed to be seen in the DNS system, or at the root or at any level. So it is an old story. And this is a slide back in the 2006, sorry, in the Wired, that says that only routers actually leak some information about the usage. And so we don't really claim that we have seen this for the first, as the first people to do that, but we won't study how this phenomenon is [inaudible] today.

So, the first part of my study is going to be looking at the leakage of ONION at A and J. Initially, this is the data that we had, but then we realize we might as well try to extrapolate from the results from the data that we had access to.

So the first data set that we have consists of six month capture from A and J, and for the ONION that it had, it had 27.6 million requests. For more than 80,000 second level domain names, coded domain names. They are basically hidden services in the TOR. And they are coming from about 170,000 [inaudible] IP addresses, coming from more than 100,000 slash 24 addresses, representing 21, or coming from 21,000 ASNs.

If you are familiar with the [inaudible] …Internet, this is a great usage of the use [inaudible]. In particular, .ONION ranked 461 of a large number of SLDs that are seen at the root, including ones that are delegated SLDs. If you look at the upper chart, which captures the volume of the number of the requests, you see there is like a steady increasing trend for the number of queries.

We wanted to understand how to present our ANJ, or for the overall traffic that could be seen at other roots. And while this slide does not really serve any great purpose in light of the later results, because we were able to actually find out right away, by looking a year where little actually has representation from all 13 name servers, but we could, using the undesirable mathematical tools, that NEG actually can give us up to 20% of the whole sample of queries at the root.

Here is something that is not surprising to many, is that the volume and the diversity of name services following certain heavy detailed

distribution, so we see that 90% of the SLDs have list that in requests, and 95% of the requests coming from less than 10 ASNs.

Very few SLD with large and diverse traffic patterns, in other words.

We here, look at some of the popular services that are listed. We put them in certain classes. So they are like larger numbers. They are 81,000 SLDs, so it's really not possible to look at them individually, and try to find out what they are. But we looked at the [head of the tail?], a head of distribution, not a tail distribution. The [inaudible] were actually has most of the requests.

And we can classify them into certain classes. They are trackers, they are deep web services. And the trackers include peer to peer systems, TOR directory services, search engines that are potentially blocked in several countries, that would be accessed through the TOR network, and other TOR related websites for the deep web. We have seen some TLD, some SLDs for the [inaudible] and the [inaudible] market list, which is another deep website.

Bitcoin is not to be confused with those classes, but we have seen some of the services with Bitcoin. Here is a listing of the top 10 of those SLDs, and the top [inaudible] with them, we find some of those [strikers], as much as 26% of the whole traffic, and the top 10 collectively have roughly 40% of the traffic. Something that is surprising but is useful to you here.

Another thing that we looked at is the [inaudible] diversity of Internet services, and here are the countries with their portion of the traffic observed for .ONION A and J. And I highlight here a few countries,

particular the United States, DE, FR, France, and Spain. And the one, the US marked in the green, is a country where we see, at MNJ we see a larger amount of traffic then you would expect for the TOR.

As reporting by TOR, the US has only 13.4% of the overall TOR, however, we can see 35.7% here. Other countries have less than their real reported usage of TOR. Germany 8.8% where we see in our study, 5.3%. So why we actually can explain why the US has more traffic here, we really not find any way to, why Germany has list traffic, except proxy. So if we look closer at where this traffic is coming from, at the right side of the slides, we find that the top contributor traffic is Google, particularly their open public DNS service.

And it's not surprising, a lot of people who use a TOR would configure. So this could be like a reason why the US has an inflated [inaudible] traffic from it.

We look at even correlations, and TOR is mainly used for circumventing censorship. And what we find is that in the, for the volume of traffic that we study, we find that there are a lot of spikes for the volume traffic that is observed at the roots. And we were able to correlate those spikes to some publically reported events. The table here shows, like the first spike corresponds to a time where [inaudible] reports on [inaudible], and this could have triggered users to intentionally try to put the SLD in their browsers perhaps, that is not properly configured, then you see that the DNS root.

Other events are highlighted here. We looked at particular event that coincide with the timeframe that we captured. In particular, we looked at the incidents in Turkey where the government blocked certain

website, such as US based websites. And we found that the timeframe that correspond to this censorship actually is, correspond some spikes around February and April.

So far we have looked at the threads or state of .ONION, and the A and J root servers, now we move on talk about that in detail, that I said. And [inaudible] we were able to analyze the same aspects that we analyzed for A and J, but over the seven years that are logged in the [inaudible]. So, as of yesterday, as it was mentioned yesterday, the [inaudible] does not have traffic logs for all root servers at all times, so the next slide will show like, where we had [inaudible] for which servers, but the overall traffic that we have seen over this seven years, comes from about five million IP addresses that are, and about 300,000 slash 24 addresses, and those queried about 18,000 unique .ONION SOD's.

Here is the list of the years that [inaudible] root, and the particular root name servers that are present in the [inaudible], and total number of queries that are observed every year. We see an overall trend, a growing trend of the number of queries observed at the root, and almost always, accept the year 2010, and we will come to explain why we see that.

However, the trend, it's a consistent trend that is growing, that we see many .ONIONs at the root. Highlighting that this could potentially be a problem and we will come to that in a second.

So, here is, the point I wanted to make here is, like how it representative are root A and J for the overall traffic that we see. And we [inaudible] where all of the roots are actually present. We find out the proportion of traffic that is shared with each root, and we find that

A and J combined are roughly about 20%. 20% of the traffic on which we based the earlier study, is reasonably sizable sample, from which we can draw conclusions about trend of growth.

So here is the number of queries and the number of SLDs that are seen each year. And in this slide, I'll try to explain why we see this spike at 2010. If you look at the lower part of the figure, in 2010, we still have a smaller number of SLDs then, say for example, 2000, however the number of queries that we see in 2010, is larger than the number of queries that we see in 2011. And overall, trend of growth is more consistent, more consistently observed in the number of SLDs rather than the number of queries.

And by digging deeper into the data set that we had, the [inaudible] data set, we observed that in 2010, certain .ONION SLD was posted publically, that triggered a lot of queries, potentially by not understanding how they should be routed. Like opening a browser that is not configured correctly with TOR bundle, that is supposed to actually route anything with TOR within the browser, in the browser within the TOR network, all of them are exposed to the [inaudible].

Here is a similar friend as [inaudible] with the exception of 2010, but here are the number of IP addresses in the [inaudible] scale, and the number of slash 24 addresses on the scale as well. So this lead us to potential causes of leakage and potential remedies for it.

So, while we don't have anything conclusive, with respect to the root causes for this, however we believe that much of it is due to ignorance of the crowd. So in order to have everything routed within the TOR network, and not be leaked to the DNS system, you have to have a

bundle installed and configured well with respect to the browser. And if you are to route something within the TOR network, you have to use the browser for which the bundle is installed.

Likely, a mistake done by many is that they would try to access the hidden service from a different browser that doesn't have the bundle for it. Another possible explanation is that [inaudible] list processing, where .ONION it serve [inaudible] in a longer query, and the browser just misunderstands as a valid DNS lookup.

A browser refreshing could be another possible explanation, the rise of malware and the usage of TOR is another good candidate and explanation. Something that the TOR team disagrees with us on is bundle misconfiguration. This could be a case for the earlier releases of TOR, but today, the bundle configuration itself is not automatically could be a contributor for the earlier observations of .ONION, but not for the recent ones.

But they shall remedy for this, well let's start with the easier one, which is just notify the user that something wrong has happened. And this is really of TOR or the browser, or wherever this traffic is coming from. Made the configuration and this seems to be something that TOR project has already done. However, something that is more DNS oriented is enable blocking of such non-delegated TLDs at the start [inaudible] resolvers.

Now we look at the potential implications for users in this narrow scope of ONION routing. So the potential implications depends on who is querying, or in particular, what is [inaudible] address I see as either a resolver or a root server. So if you visualize the address then, this is

perhaps the most severe condition, and if say for example, a censer, a government [inaudible] censer, is seeing some .ONION queries, he would not like come to the user and ask him, "Oh, your browser is broken, or your bundle is broken."

That would not really be a question. So, there is a clear identification here, and that identification can be done by many. If the IP address is a recursive IP address for the outbound queries, there is no problem because most of those, well the majority of those outbound queries are aggregated, and lists, it's a list of a [inaudible] of IP addresses of individual user is not exposed.

However, even if we look at different types of recursive, then the censors may prevent those recursive from sharing such information with potential [inaudible]. So if you look at ISP resolvers, it's still the same case that outbound queries are aggregated, and incentive may guard user privacy because users are being ISPs basically for curing their traffic directly.

However, if you look at all of the resolvers, outbound queries are aggregated, but some [inaudible] privacy since, those resolvers capitalize on the data they observe from those users. And this is a nice place to plug, the current efforts are going in the community. So deprive DNS private exchange, and privacy enhanced resolutions could be utilized in this domain to address the problem of the global observers and remedy the risk for the users.

In conclusion, we have looked at .ONION at the DNS roots. We have examined some of the characteristics that they have over a period of time, and we have highlighted that there is an increased traffic spikes

with certain events that are [inaudible]. Certain causes of the leak are unclear to us, so we invite the community to study those if anybody is interested, or we highlight some potential causes including misconfiguration, search lists, so on and so forth.

We plan to continue examining the leak and understand its real causes, and look into other non-delegated privacy related TLDs. Thank you very much and if you have any questions, I'll be happy to answer it.

[APPLAUSE]

UNIDENTIFIED SPEAKER: Warren [inaudible] is first.

WARREN: Hey. Warren [inaudible] Google. More of a comment than a question. Myself and Andrew Sullivan actually have a draft that sort of mentions how you can address some of this. We're suggesting that there be one TLD reserved for all of these sort of DNS like names in a non-DNS context.

AZIZ MOHAISEN: I read that.

WARREN: Okay, cool. And if that was forward, I think it would be useful for this, and [inaudible] and all of the other…

AZIZ MOHAISEN: Definitely, I agree with that.

UNIDENTIFIED SPEAKER: Okay. Thank you very much. That was a very interesting presentation.

[APPLAUSE]

UNIDENTIFIED SPEAKER: Next will be Peter Marx. He is the Chief Technology Officer of the Municipality of City of Council, I don't know what they call it in the US, of Los Angeles. It's a big enterprise. I am quite sure he has got a few things to say. [Inaudible]. How do we organize the presentation?

It's organized, don't worry.

All right. What we usually do is we ask the host, this is usually the ccTLD, to make a host presentation on their setup and things. This time I understand there is not really a host. He is going to use his own. He's using his own presentation.

He wants to use his own. Okay.

And so I'm actually quite happy that we found a local host who can speak to us. I'm just trying to make small talk until you get these things working. All right.

PETER MARX: Okay. Can everybody hear me? So, there is actually a city here, it's a municipality, it's called the City of Los Angeles. I think probably everyone here has heard of it, yes? Just want to make sure. Because *The Guardian* newspaper a few months ago actually did a study of brands for large cities, and LA for whatever reason, ended up as being the top city brand in the world. Who knew?

And so I was asked by ICANN if I could do a quick presentation, so I only brought 62 slides. [Laughter] No, I'm kidding. And that I would actually take a look, see if we took a step back… I've been writing software for my entire professional career, in fact, before I had a career. But maybe we take a step back and frame the conversation and actually take a look at the city around you.

And how the Internet is slowly beginning to affect changes in the city, because I think as an organization, as a group of engineers, as a group of technologists, you will all see the implications and lots, and lots of opportunities and challenges to work on. By way of background, I'm the Chief Innovation and Technology Officer for the City of Los Angeles.

I took this job about eight months ago. Before that, I handled corporate and strategy for a little chip company called [inaudible], which many of you will have heard of. How many people here have smartphones? Anybody willing to admit that you don't? Okay, good. Because they'll be prominent in all of this.

But let's start in the past. So you're sitting on the edge of a continent. The only reason why this city exists is because the railroad ended up here, and in many ways, as the far left end of North America, you know, the mayor likes to refer to Los Angeles as being the northern capital, if

you will, of Latin America, the eastern capital of Asia, and the western capital of the United States, because it is the nexus of so much that comes through here.

And in the 20's with oil and automobiles, you know, this city is quite famous for, well, we can leave it at traffic, of the automobile type, and we'll talk a little bit more about that. In 1924 they actually started creating the first traffic management system. And this is well before routers, well before other types of traffic management. So of the electronic type, we're talking about trying to manage machines being driven by humans.

And, as you go further, I think most of us know that the [inaudible], one of the original nodes on the [Arp Net] was at UCLA, which is merely, I'll say, three kilometers away. And as such, you know, Los Angeles has had a long history of dealing with things like the Internet, and cellular, and digital communications, and so on, for a long time, this has been the top cellular market in the United States. And it has really got the highest tariff trades.

And if we go back to traffic management systems, you know, with the 1984 Olympics, the city decided to go and actually start to put in a digital, electronic, traffic management system, underneath all the streets in Los Angeles. And at the moment, it's actually now fully built out. There are 4,500 intersections, which have stop lights. Something like 55,000 or more traffic censors from around the city.

And they report the traffic back, once a second, to computers downtown. And the week before last, we actually put out a request for information, basically a public tender, asking people, all of you, to come

tell us what you would do if you had access to that real time traffic management system.

And by the way, of course the answers are for the public good, I just want to point that out. We don't want to… We've seen proposals of people doing public operas, using automobiles, we don't want to see more operas, although they're interesting, we'd rather see better traffic flow.

In addition, we've also have real time APIs to things like the parking meters. You may think about it, but 40% of every trip in an automobile is usually spent looking for parking. And if you think about it, the issue there is discovering an open parking space. Anybody here ever frive around looking for a parking space?

It's one of those. And so the question becomes, how do we actually take so much of the lessons from the Internet, of things like discovery, and navigability, and user experience, and actually apply it to this really mundane day to day experience of driving around.

Because here in Los Angeles, we're sort of famous for how we move. And I apologize for the low resolution of this. I'm happy to share high resolution with anybody who wants them. These aren't published, they're fun. But here in LA, roughly three quarters of the people, two-thirds of the people, drive themselves, in a machine, by themselves to work.

We have five subway lines under construction now. In Los Angeles, there is more public transportation available, than anywhere else in the country. And once more, there is more public transportation under

construction here than anywhere else. In fact, there is a big fight, well let me avoid that word.

There is a movement, there is a subway that is actually coming to right here. It's called the Purple Line, and it will arrive in a few years, and it will arrive literally, as I recall, directly behind this building. On its way, as it meanders through up to UCLA and then beyond. There is another subway line to the south of us called the Expo line, which is jokingly referred to as the [Text-po] line, that is going to open up downtown Los Angeles to the ocean.

And we're seeing a tremendous amount of start-up activity happening around these transit corridors, not surprisingly. To continue the story, most popular car in California is, of course, hybrids, Prius. You know, the aforementioned smartphone showed up here.

I did a lot of work in Asia in my life, and we're all very familiar with how far behind the US is in terms of, you know, usually adopting cellular technology, we've obviously taken to it with an incredible enthusiasm. And then another thing happened very quietly. If you go out on the streets, and you look at the lights at the top of the poles, 30 feet above your head, 270,000 have been replaced with LEDs, across the city of Los Angeles.

It's one of the biggest green projects every done. And many of them are mesh networked together for remote monitoring and maintenance. And if you think about a streetlight, a streetlight makes a really excellent place for things like public Wi-Fi, makes a really wonderful place for things like small cells. In fact, the Phillips, and there are other brands out there, you know, are now beginning to promote products where

you can actually put in a couple of small cell units into a street light, and turn it into a public utility. More on that to come.

In LA, the car that I drove over here, which is an electric Fiat, is less expensive to me on a monthly basis then my smartphone. It's kind of remarkable, and we're beginning to see a lot of electric vehicles running around the city of Los Angeles. So much so, that we're actually beginning to put electric vehicle charging stations on the side of streetlights, on public parking spaces.

And the idea being that you can start to deliver services, you know, literally off of the municipality. And a lot of this comes back to communications, how do you pay for it? How do you recognize it? How do you understand proximity? How do you understand authentication? Etc. Things like that. And then, of course, we're a little bit later to the game, after the UK, and Japan, and so on, but a few months ago, we turned on their first trial of 52,000 smart meters for electric power monitoring.

Because the city of LA is both larger than you think and smaller than you think. We have the largest municipal water and power company in the country, the department of water and power is quite famous. If you ever watched the movie *Chinatown*, it's well worth a look to understand how important water is to California.

The port of Los Angeles is also operated by the city. 44% of the US economy goes through the ports here. Think about the number of different items and, you know, the logistics of such a thing. The airport, LAX, which probably many of you came through, unless you came through the port on a ship, but that's probably a little unlikely.

If you came in LAX, something like 64 million people arrived at LAX every single year. And so the city operates these very, very large infrastructure pieces. And then, of course, it also operates a street system, this is to go back to the real time traffic system, that spans 470 square miles, and my head is not good enough, right now, to translate that to kilometers, but somebody can do it here.

But here is the thing, it's divided by a mountain range, in the middle, and it has a vast geographic spread. Now different then, let's say the east coast of the United States, the city of Los Angeles is one of 88 different cities in the county of Los Angeles. There are many overlapping governments and entities around here. So coordinating digital infrastructure across the geography, is actually quite a difficult task for a lot of social and governmental reasons.

However, within the city, we do have the saying, which is that we operate a city that is, you know, fairly significant in stature. It's the number two city in the country. And where I think it gets interesting is that, I took a slide from the Internet of [things] presentation by Cisco. Just for fun. And what you'll notice about it, it's a typical slide.

You know, we've all seen the Internet of things, presentations being, you know, my alarm clock wakes me up, it understands how well I slept, maybe gives me a couple of extra minutes of sleep, because not only do I need it, I always need it, but the train is running a little bit late, and then the connection needs to change, and you know, so forth and so on. Right?

Well, if you start to think about how these things interoperate with the public infrastructure, it gets kind of interesting. Because we tend to

think of these devices as being very much on their own. I started a project at [Call com] called [Gimble], which is around context awareness. The idea that a smartphone could understand your context, whether indoors or outdoors, whether you're walking into a place, whether you're late into a meeting.

You know, based upon the data that was available to the phone, and the censor readings that are available to the phone. But the fact of the matter is that, interoperability with the infrastructure around you is probably a lot more valuable. And so, when you get into vehicles, how many people here are working on things like…

You know, here in the US, we talk about DSRC. Anybody here working on vehicle infrastructure? So there is a lot of work that's happening around by entities who are focused on, how does a car interoperate with the stoplight? I mention that we have 4,500 stoplights that are managed via computer, well those stoplights, they show three lights. They're for human beings. Right?

Where a human sees a red, green, or yellow light, and then reacts, you know, appropriately, hopefully appropriately. Well when you start to get into autonomous vehicles, when you start to get into connected vehicles, you know, we now need a way for those vehicles, those robots, to actually interoperate with the infrastructure around it. And the engineering challenges are pretty significant around them.

We do have connected vehicles today. All of the public buses, all of the subways, all of the, you know, those public vehicles have vehicle locating systems, and they interoperate with the infrastructure at least to a light degree. I'll tell you a little trick, if you ever want to get from

one side of Los Angeles to the other, may not be the fastest, but it will be one of the most reliable ways, is to follow those big red buses, the rapid buses, that run around the streets, because the intersections will adjust the timing to keep those buses on schedule.

And it's very important. In fact, we license out our traffic management algorithms to other cities, because the estimate is that it saves us probably with 16% travel time, across the city. Now, for all of us engineers, proving that is actually a little bit difficult because you have to turn it off, in order to have a real AB test, and nobody wants to go there.

But when you start to think about the Internet of things, the interoperability in the systems, the governance, the protocols, the standards, the customs, that are going to be required for lots and lots of different devices to interoperate with, let's say, an urban infrastructure, especially one as large as Los Angeles, it gets to be really quite something.

And some of the challenges and opportunity, you know, quite frankly, I'll speak a little bit about it, come down to people and culture. I think we as technologists tend to sort of forget the people aspect of things. This is my theory. But there is a lot of work that has to be done in that space.

Here in LA, despite the fact that this is one of the original three nodes on the [inaudible], the original three nodes on the Internet, we have a problem that we have, you know, many hundreds of thousands of people, who are not on the Internet today. They're on the other side of the digital divide.

And I think that as you go across the world, we see that same issue. Not everybody is carrying a smartphone, or has access to the online world. This audience, all of you, you have access, but within just mere miles, kilometers of here, there are a lot of people who don't. And we have a cultural issue as we try to get them onto the nodes.

Identity, you know the classic one, authentication, classic one. Payments. We're beginning to see a lot of activity on the payment side, in secure payments and all of the rest of it. Large brands like Apple, for example, you know, have recently made mobile and online payments, you know, take a step up, at least in theory, but if you start to think about your daily life, one of the most common sets of payments that many people do, are for things like getting onto the bus or to the subway.

Or, for that matter, paying for parking. Or just whatever, your library card. We have lots and lots of transactions that happen every single day. And mobile payments is something that would really help us because we would like to have mobile payments so that we can incentive people so that two-thirds of the people who are driving solo in their car, maybe we can give them incentives to actually take public transportation.

After all, we're putting billions of dollars into digging holes into the ground and running trains around, and we would like people to use it because we have a sustainability thing, you know, a sustainability goal across the city. And then, I think we all understand that on the Internet of things…

As a friend of mine said, when I started [inaudible], it was M to M, right? Machine to machine. And then one day somebody started calling it the Internet of things. And then, one of the SVP at [inaudible] said, "No, it's the Internet of everything." And then one of the marketing people said, "If you call the Internet of everything, that doesn't leave any room for anyone else."

But that's a separate conversation. But no matter what, we're walking into a world where there is a tremendous number of connected things. And I'll go back to how the city manages it. We don't know, I don't think any of us in this room know, what will happen when we start running a lot of, you know, autonomous, semi-autonomous vehicles running around the streets, whether they're drones, whether they're FedEx, whether they're buses, or whatever.

You know, how do we actually manage where public safety? How do we manage for user experience? And things like that. And that's going to require both the local governments, and the standards organizations, and all of you, and all of the private companies to come up for some system. We're years away from that, I think. Most people focus again, on the device, and don't focus, I think ICANN is obviously well positioned to think about the infrastructure required, but most people tend to think, "Hey, I've got this device. And I'm not thinking about all of the invisible web of stuff behind it."

To a greater degree, you know, there is obviously a big fight going around things like Net Neutrality. In Los Angeles, for the Federal Communications Commission, we have filed position statements, letters, on Network Neutrality. We have filed, by the way, we're for it,

of course. Because we have a creative industry here, that depends upon the Internet for distribution of all of its output, whether it be music, or video games, film, or TV.

And then in addition, we have a whole bunch of stuff going on between the debates about license and un-licensed spectrum. You know, Comcast is buying Time Warner Cable, two companies here in the US. AT&T is buying DirectTV. These have profound implications for the type of infrastructure that we see inside of North American cities.

And then I'm always going to come back to it, security and privacy. I have yet to see anybody navigate both security and privacy, especially recently, in a perfect way. And these are getting more and more into the realm of being issues that really have to be dealt with. I don't know how, but you're all going to have to deal with it.

We just watched a talk on, you know, ONION DNS and everything else, but it goes so much further than that. Personally identifiable information. What happens when a foreign, well what of you? When you come to the United States, how is your information handled? Because quite frankly, we have different laws and different rules and different customs around privacy and security everywhere.

And so, electrons know no boundaries, right? But to come back to the culture. LA has got a very big, complicated culture. There is a street, I mentioned at Wilshire Boulevard, and somebody did this kind of fun infographic, but Los Angeles is the number one second population, and I can't remember whether it's for nine countries, or 11 countries, meaning that there are in LA alone, there are the second largest

number of people for another country living in one place, whether it be, you know…

I'll leave you to go fact check it, but it's quite something. We have an incredible number of languages, voices, cultures, people and so on. And along one street, if you start to think about, you know, how do you run robots and autonomous vehicles around one particular street, Wilshire is a good example of the complexity that you're going to run into. The number of businesses, you know, the number of different types of entities that actually exist around here. And so, I'm just going to spend a couple of moments focused on user experience. I think we all view, well this is my personal view. And you'll see in these photographs, that there is a common element, and it's a mobile device, but when it comes to explaining what it is that you're standing in front of, when it comes to, you know, improving the user experience of the city, we're at this moment, right now, where the Internet, plus wireless, plus intelligent devices, is now able to go and bring us a different user experience of actually being in a city.

So if you're standing in front of a statute, this is one in front of City Hall, I never knew who this fellow was. But you can look him up on Wikipedia. And when you walk into City Hall downtown, so it's a famous old building. I invite you to go there, it has got great views, it's not exactly setup to be self-explanatory. Yet, your mobile device is a natural [inaudible], right? And similarly as you're walking around the streets, especially for many of you at ICANN who are new to the city, may or may never have been here before, but simply understanding public safety, how to get onto a bus, how to pay for a parking meter, the mobile device really represents an opportunity to go and really tackle

the user experience, in a cross-cultural, multi-lingual, you know, easy age, non-dependent way.

And very few cities are doing this right now, probably Hong Kong with the Octopus card, is probably the best example, but here in LA, we really, as a focus, would like to make the user experience of being around the city better, so that we avoid things like this, where you run into construction sites, un-expectantly.

Now, there is a human element behind this, by the way, this construction site is right in front of the Department of Transportation. And I will let you all ponder the significance of that. But the only way that you will know about it is to actually drive into the construction zone. And usually, as we all know, all over the world, if you start seeing signs saying, construction ahead, usually you're already in the traffic caused by the construction, and so that's an issue for all of us.

It's a poor user experience, to discover the problem by feeling around, shall we say. It's a worse experience that here in Los Angeles, we issue parking citations for poorly parked cars. That's a poor user experience. It gets worse when we tow your car. It gets worse when, to find your towed car, which usually happens after about, well 6:00 in the evening, we turned off the public information service at 4:30, because that's when they go home, so you can't find your car after we've ticketed your car, towed your car, and taken it away.

And on the list of user experiences, these are not good. I think I'm sort of stating the obvious. So towards that end, what we have done is we've joined cities around the world have done this, and governments around the world have done this, but we've joined a movement called

the open data movement, and we have put up, at the end of May, beginning of June, we put up a website called data dot LA city dot org.

And we invite everybody to use it, because we publish as many datasets as possible on the actual city, of everything. Everything we can find. And slowly but surely, we're taking all of the data that's available from the city government, and also from the county above us, and the state above us, and the Federal Government above us, we're publishing that data in a regular, updated way, and sometimes in real time.

So that people can actually go off and use it, and I'll come back to that request for information that, you know, how would you want to interact with the public stoplights? It's kind of crazy, right? But here is another part of that. In so doing, it allows us to do things, last week, I guess it was the week before last, we announced, and New York did it just before us, we announced that we have a goal, at the city, at the Department of Transportation, to get to zero pedestrian and bicycle fatalities.

There is a lot of ways to it. Right now, it's complicated, you know. A lot of bicycle injuries don't get reported. And so it has to be a concerted effort of not only the public safety folks, the police, and fire, but also the emergency rooms, also the bicyclists and the pedestrians themselves, also the city government, also Department of Transportation, lots of other folks, have to get together.

It's only possible because of the net, obviously, to be able to begin to do this. And to do it in a real time way, it's kind of fascinating. But I will point out that companies like Honda Automobiles from Japan, made it a

statement of theirs several years ago, that they wanted to produce a car that could not kill somebody, could not kill a pedestrian.

It's really important. And we think that cities and urban infrastructure have a big part to plan this, but they could not have played a part in this before the net, and before wireless, and before all of this stuff that we're doing here. And, as you go around, there is lots and lots and lots of different types of things that we focus on, whether it's, you know, here again in Los Angeles, it's about livability, public safety, it's about water, it's about energy, it's about air quality, quite famously so, and so on and so forth.

I mean, if you look across the city, there are a number of different APIs that we are making available, in order for the outside world, whether it's public or private, academic or individual, to be able to interoperate with us. And with that, thank you very much for your time.

[APPLAUSE]

UNIDENTIFIED SPEAKER: Thank you very much. Me coming from a small town, of a developing country, I'm happy if I'm able to just pay my bills, and the computer is working out the banker's working, which in our town, is very rarely happening. Quite interesting. Any questions?

PETER MARX: There has got to be one. There must be one.

Oh yes.

UNIDENTIFIED SPEAKER: I come from the city of Santiago in Chile, where traffic, I guess, is much crazier than it is here. And many people down there use a service called [ways?], that does wonderful, in terms of giving you the best possible route that it knows, and without the need to establish a public infrastructure, it collects information in real time from thousands of drivers that are using the system and providing it with information.

It would be great if such a huge amount of information could be integrated with other services, but I don't know if there is a way to do that since that is proprietary information, and I don't know if you think, I thought about something with that kind of service.

PETER MARX: We announced, I think it was last week, it might have been two weeks ago, we announced that we were working with [Ways], and Google, and so forth, around sharing, better sharing of information like public transportation, bus schedules, and things like that. But I will invite you to think of the following, which is that [Ways] routing algorithms are based upon, well obviously, geo looking information like streets and so on, and then crowd source traffic information.

I use [Ways] every day. I think like, how many people here use [Ways]? Come on. Yeah, you see? But one of the issues with [Ways], is that [Ways] is increasingly routing cars through small streets. They're routing cars and vehicles through a transportation system that isn't necessarily designed to take them.

And once more, [Ways] is also doing it without any understanding of whether, for example, what the speed is that the traffic lights are synched to. And we think that, if you were to provide [Ways], and Google, and the other routing providers, of which there are many, with better information about how the system is working, that they will be able to do better routing, not only for better user experience, but also for better safety.

So I think [Ways] is great. I think there is a function that they don't have, just off of crowd sourcing, that can make their service better.

UNIDENTIFIED SPEAKER: If you provide with that information to [Ways], you'll be getting, or return some information from it?

PETER MARX: I wouldn't want to speculate, but I would certainly expect and want their help in driving to the public good. We want to know about problems, we want to know about how to make a better city. You know, all for the public good. So, yes sir.

WARREN: Warren [inaudible], Google. Not talking about the [Ways] stuff. So you mentioned the whole digital divide, not everybody having access to the Internet, blah, blah, blah. You also were talking about, you've just gone through and replaced a whole bunch of streetlights with LEDs, etc. What are you doing to get it so that people can get access to those

poles? And access points, etc. so that everyone else can get on the Internet?

PETER MARX: So we put out a request for information for this thing called the LA Community Broadband Network. And what it was, was we actually listed all of our assets across the city. All of the fiber, all of the city owned assets and everything else for communication providers to be able to come in and say, "We would be able to do X, Y, and Z."

Of which, Google, by the way, is a very famous participant, you know, at the higher levels with things like Google Fiber. And our expectation is that that will end up as being a RFP, a request for proposal that will go out. But one of the clearest things that we can do is that, right now, the city requires a permit for anybody to build anything. And we want to make those permits for communications as lightweight as possible.

Obviously, we have to balance community, and safety, and everything else, but one of the clearest things that we want to be able to do is to allow communications providers to efficiently be able to drive high-end, communications infrastructure across the city, because apparently 20% of the total cost of deploying broadband infrastructure is permits. So does that help?

Oh no, no. It's just the beginning annoyance. I'm sure there is more. Any other questions?

UNIDENTIFIED SPEAKER: This is going to be the last question. [Inaudible]. One of the things we found in the conventional Internet, is low cost, high numbered boxes that are deployed at consumer sites, have turned toxic on us, and actually been used in massive denial of service attacks, like open DNS resolvers and so on.

When you talk about a city that is replete with censors and digital infrastructure, which is kind of this larger vision of a massive deployment, of low cost, ubiquitous, and very cheap bits of infrastructure, how are we going to ensure that this operates with integrity? What sorts of things and issues do you find even now, with a toxic network where this stuff gets turned against us, and used in terms of denial of services and similar?

PETER MARX: Well, you know, it's a great question. I'll say in the following way, which is, I don't think any city or any entity has got this down. Just, and by the way, I don't think it's limited to low cost devices as being the only issue. There are some very expensive devices that also are usable for these sorts of problems.

And so, we did set up a cyber-intrusion command center. A CICC, which for the first time, brings together a whole bunch of different entities around the city. Everything from the federal level, secret service and FBI, to the city level, you know, port police, LAX, you know, infrastructure from the Department of Water and Power, think of all of those status systems, and Department of Transportation, everything else.

And so, we're really working because not only is there cyber-security, but here in Los Angeles, we have other issues, like well, there are very famous earthquakes that happen. They look a lot like a denial of service attack, in certain physical ways, right? And we're, I wouldn't pretend to say that we're smarter than anyone else, but we're certainly working pretty hard on all of this, and we're certainly all ears to what people have to say.

The Internet of things is either going to open a lot of doors, and I mean literally, you could use the thing to open a door, or it's going to, you know, people talk about breaking their phone, people talk about breaking their house, you know, I'd hate to see where that progression could go. Thank you. Thank you very much.

[APPLAUSE]

UNIDENTIFIED SPEAKER: Again, thank you very much. Next one is Patrik Fältström. Patrik Fältström is our IETF liaison this week, this time around.

PATRIK FÄLTSTRÖM: Thank you very much. I have one question first, how much time do I have? I must admit that I don't really remember.

UNIDENTIFIED SPEAKER: 20 minutes.

PATRIK FÄLTSTRÖM: 20 minutes. Okay. Good.

UNIDENTIFIED SPEAKER: If you could do it shorter, everybody will go earlier to lunch.

PATRIK FÄLTSTRÖM: I haven't had lunch myself, so this is cool. Okay, so what I was asked to do is to bring an update on where we are in the nationalized domain names. And one of the reasons is that, there seems to be some confusion in the community what kind of changes there are between the 2003 and 2008 version. This slide deck was something that I whipped together last night, during the night, this morning, and consistent of what I have as material, and I will not go through everything, but I hope the complete material will be something that helps you.

So please reach out to me if it is the case that you have any questions. First, what we to do regarding internationalized domain names? Well, we decided a long, long time ago in a different galaxy, or something, that this is something that could create some issues, if just add non-ASCII characters to what we call host names.

And even though the DNS protocol itself could handle, of course, any kind of value in the bytes, in the DNS packet, that is not the case when we talk about the applications. It was also the case that we were a little bit nervous, and we wanted to be a little bit conservative, so it was the case that the encoding that we did was wrong, we didn't want to destroy the complete domain name namespace, so at the end of the day, we found that using a prefix, x m dash dash, was good because if it

was the case that we screwed up, which we still don't know whether we did, we can create another prefix and start all over again.

Of course, no one really wanted to do that, but at least that's a possibility. The good things with the x m dash dash thing is that what we are passing around in application delay protocols, etc. is only ASCII. We know that all the applications can handle that. Yes, the string ends up being a little bit longer, it looks ugly, there is no support for it, but on the other hand, if someone has, for example, an email client, and get an email with an internationalized domain name in it, in the address, it is possible to reply to the email.

So the ability to reply, for example, to, it inclines that you're not understand IDN was really, really important for us. We do see IDN in other places, specific being at the ICANN meeting, we see that the whole issue with the internationalized domain names need to be taken care of, not only in the standards themselves, but also, and also in applications, but also in various policies. What kind of domain names would you be able to register, what characters should be allowed in domain names?

What characters should be allowed in applications? And this is something which I'm trying to concentrate on in this presentation, because it seems to be the case that people do believe that, well, as soon as we have non-ASCII, then we can allow any Unicode character whatsoever. That is not true. There are many Unicode characters that are actually not found at all, to try to use in a display environment, or add it, or take care of.

Those of you who have tried to write a text editor, or parser can sort of support that. So, there are multiple issues with just using Unicode, so we need to use a subset of Unicode. So for example, one of the issues that we have is that, I want to register domain name, and I register my last name, Fältström dot SA, which is the name that I registered. Other people are typing in a domain name and do that in capital letters, in Swedish the uppercase version looks like this.

Of course, the matching rules that we use in DNS say that only ASCII characters, like A to Z, are to be matched in the case in a sensitive manner. So the question is then, how do you do the matching of the characters in this domain name, that is not A to Z?

Well, we can do case insensitive matching, people say very neatly, but the problem is that in Unicode, the case insensitivity is not defined. So that is just not the case, that it's poorly implemented, it does not exist. What you can do though, is that you can case fold the stream, either to lower case or upper case, and then do the matching. And you should be aware that there are some characters which case fold, for example, to lower case, and then you move it to upper case, and unfortunately those two functions are not yet inverse of each other, as it is in A to Z.

So, that is just one example where you might end up in problems. Of course, in some scripts, for example, a [rabbit?] script, you actually use, you have different display and of the characters when you swap between right to left, and left to right, and where the characters are in the beginning and in the middle of the words, etc. etc.

So there are like multiple [dragons?] in here, and still we need to make sure that, as much as possible, is usable in the DNS and domain names.

So if you look at it more closely on the actual flow of domain names, this is one picture. Each one of us in this room probably have our own picture, this is my picture, and the problem I have, I have tried to look at when I was doing the IDN standard, is that where the registrant, that is passing a domain name in two different directions, to the register, that passed to registry, that pass it to a zone, and then it also passes the same thing to some DNS operator, or into the software they're using, and things end up in a zone file, or in a name server, and then you have resolvers and other, sorry, recursive resolvers and caches all over the place.

And then I have my, a friend of mine that want to use this domain name, or customers, or potential customers, and they entered whatever they think the domain name is in an application, and the resolver sends it over, and then in some magic way, they're supposed to do the matching here. The match that is equivalent or not equivalent between what I have typed in as a registrant, or what my friend has typed in.

And you should all know that there are people that have PhD degrees on how to do matching in Unicode. Okay? This is really, really complicated. And it's even more complicated to try to do it without, for example, knowing what language we're talking about. So matching [inaudible] or language and context specific. And of course, in DNS, like we don't know what context a certain string when we've got a DNS packet in there.

I don't know how the application is supposed to use that. So what we had to come up with, was some kind of generic algorithm for doing matching. Okay. Fast forward. There was some conclusions, okay, we

cannot change the matching algorithm in servers, so we use ASCII all over the place. We do the IDN translation between the presentation and the communication layer, we left the communication, they use ASCII just like we did before.

So for example here, we have like two different strings in the [rabbit] script, it looks like for you, and maybe specifically for us that don't read a [rabbit] like me, this looks exactly the same. Of course, those two strings are not the same. It's also the case that for a computer, it might get a case that the two strings are different, because you can either have a character, Unicode characters, 623, or you can have a combination of two other characters, and the question is, should those two be equivalent or not?

It's also the case that we have some other kind of issues, for example, with the numbers and [inaudible] which are completely different in certain scripts. Once again, the example is a non-ASCII script, so these are the eastern Arabic digits, which looks a little bit different, but it looks the same but of course they're different. And there are multiple of these various different kind of things.

So, how do we handle this? Well, if you look at what, in ICANN context, is called variants, it might be the case if you take the, on the top here, you see that someone want to register A, and in the standards says this is an A, and then we register A, everything is fine and dandy, just like with ASCI.

It might be the case that someone would want to register a character B, but character B is then not allowed, according to the IDN standard, we cannot allow that code point. It might also be the case when

registering, that someone wants to register C, the standard says C is okay to register, but it might be the case that the registry have the policy saying that, well if you register C, then you should also register A, which means you can have sort of combined, various streams can block each other because they are too similar to allow two different registrants having the same, having domain names which are sort of, in some context, should be treated as being the same.

It might also be the case, of course, that you also try to register D, the standard says D is okay to register, but the registry policy says, "No we cannot allow those kind of characters in our zone." Whoa. Okay, I should use this and go to [inaudible]. Okay. There.

Another thing to remember, which I unfortunately hear too much in the ICANN context, I was in a small session with GNSO yesterday, and people, when they speak at the microphone, they use the word language and script as they were equivalents, or they were very sloppy, to be frank, when they used the word language and when they used the word script.

When we're dealing with DNS and domain names, we talk about scripts, we don't talk about languages. Languages are something that exist in the presentation layer, within applications, nothing else, or you can have it as an extra [inaudible] data when you post around streams. Here is an example for Arabic script, where you have the Arabic language, normally only use the characters which are in green, but if you look at other languages that use Arabic, if you just, okay.

If you look also at Persian, Urdu, and Pashto, you see that there are many more characters that are in use. So another mistake that people

use is that, part of this is, of course, that for many scripts, like Arabic, you have this script name is exactly the same as a language name, and normally the dominant language for that script.

So that's why it's easy for people to make that mistake. This is the same thing for Hebrew, for example, where the Hebrew language is using actually the smallest set of the Hebrew script, compared to other languages that use the Hebrew script, like Yiddish which uses more characters from the Hebrew script than what the language Hebrew is doing.

Okay. More things. So, we need the rules, we need to agree on what code points to use, etc. So from the beginning, we had these four, RC 34 54, 34 90, 91 to 92, that talked about how to, what code points where allowed and how to use them based on Unicode 3.2. So it specified the algorithm, it specified the ASCII coding, etc.

So it had something called string prep that talked about illegal code points, it had a mapping table, etc. used for [inaudible] IDN, and other protocols as well. And here they are. So what happened was that, myself and Jon [inaudible], and some of you in the room know both of us, one or two or something.

Anyways, we were looking at this and saw that are kind of serious problems with it the existing IDN standard. So we wrote a document that, for example, said that the current standard just didn't make sense, as the Unicode released new version of the Unicode standard. The mappings didn't really make sense, because what mappings you wanted to use is actually very context dependent, so we wanted to have a specified standard for just the script themselves.

One important thing that this is talking about is that, it's real important the mapping between the Unicode version and the ASCII version of the same string, should be functions that are inverse to each other, which means that you can map back and forth between the ASCII version and the Unicode version without losing any data. That is, from my perspective, also an implement of the largest problem with the IDA 2003.

There were also some errors by directional scripts that was problematic, and also we forgot, most of the troublesome issues with non-space and code points, those are characters that are not visible or not printed at all.

So for example, there are specifically two different kind of spaces. The non-spacing space, and the spacing space, and those [inaudible]. Non-spacing space are kind of funky. So, here for example, here is another example where you have a Hebrew character, string. This was something we completely missed. We talked about a string being right to left, or left to right, depending on what directionality the first and last character has, but Hebrew there are cases where you have the lost character is actually a character which is, doesn't have any directionality, and that is the [inaudible], I'm probably pronounced the character wrong, but you see the character here below.

That's the lost character which doesn't have any directionality. I remember a note here, it's the lost character in the string, but for many people in this room, including myself, I would say this is some kind of blob that is below the first character. In Hebrew, of course, this is the lost character in the string.

Okay. Presentation layers, kind of fun. Whoa. Okay. Think about how to do selection.

UNIDENTIFIED SPEAKER: Three minutes.

PATRIK FÄLTSTRÖM: Three minutes, cool. So did we do? We came up with [inaudible] 2008. Also a few documents will not change the encoding, hooray. Backward compatible, hooray, except for a few code points, shhh, don't tell anyone.

What it does, it explicitly separates what you register in the DNS, instead of what the old IDNA standard, they said what code points to use, because there might be a mapping in between the two, so [inaudible] 2008 only talk about what you can use in, as domain names in protocols.

So how you choose to map them in application whatever, up to you as the developer. And because of that, there is an inverse functions that you label the Unicode version and the ASCII version. So we have a new set of documents, 58 9, 58 94, and we just go through them. So the most important thing is this, 58 92. Instead of listing the code points which you can use, it comes up with an algorithm that you can apply to any version of Unicode, and it would calculate to you for a given code point for a string, whether it's possible to use or not.

So when you update your Unicode library in your computers or operating systems, or whatever it is, it will, you just re-compute the, you

apply the algorithm to the new version of the Unicode and everything is fine. So it's the algorithm that is normative and the tables are non-normative.

IANA is publishing a calculated, non-normative table, which of course, if you do the calculations, you should come to the same conclusion as IANA. So there is a table that you can grab if you want to, but that one is update when IANA is releasing, sorry Unicode consortium is releasing new versions of Unicode.

So, there were mappings in [inaudible] 2003, the new standard does not include mappings. That doesn't mean that you should not do mappings, but you'll now have the freedom of doing mappings the way you want, or agree to in other standard organizations or whatever.

So there are some indications for the registries, because now a days, a registry really has to talk about what characters to use. So whether mapping should be done and really define what actually should be able to register. There are a few code points that have changed, for example the [Sharp S], but that is a change that is based on a change in the Unicode, standard because case folding or [Sharp S] was not possible earlier in 3.2, but it is case folding for [Sharp S] is okay later on.

And ITF decided to change the, remove the mapping from [Sharp S] to double S, and instead have [Sharp S] as a separate character, which allow that… There are some registries that have had transition mechanisms to allow ready strands to have [Sharp S] and double S of registration of domain names. But there are very, very few characters, I think there are three of them.

Let's see, what is important where we are now, is that Unicode has moved over time from 5.2 to 7.0. We came up with 64 52, that we're looking at the changes to 6.0, and there were three characters which had incompatible changes in the… I don't know if anyone in this room is actually using these characters in their day to day life, I doubt there are because these are very seldom used characters in very rare scripts.

So we decided in the ITF that it is more important to follow the Unicode consortium changes and be compatible with Unicode, than create our own exceptions. We were also doing an outreach of registries, and we just could not find any trace of use of these characters in any domain name on the Internet.

So the ITF made a decision to not be backward compatible, so if you use Unicode 6 instead of 5.2, you will get different results on these code points. The last thing is that these three code points move from not being allowed to being allowed, which is also something that is, it's easy to go in that direction than the other direction when you upgrade Unicode tables.

Unicode consortium, I want to mention this, came up with technical recommendation 46, that talk about how to do transitions in the application from [inaudible] 2003 to [inaudible] 2008. My personal perspective, is that unfortunately it doesn't really talk about how to complete the transition, it talks about sort of running both at the same time.

Personally I think it's, the faster you can move to really use 2008, the better. But TR 46 is talking about the various compatibilities in a really good way, but please do the transition and finish that. So you should

use 2008. The last thing, you're only like two minutes late, is that we do have a draft in IETF at the moment, because there is one code point that made us, that triggered something that we might have missed.

And that is in Unicode 7.0. We have an Arabic letter, which is sort of a combination of two earlier allowed code points, and the question is whether it is something that we should allow or not allow, and this unfortunately have some implication…

The answer to this question depends on whether, for example, you're talk about the Arabic language used, or the Farsi use of the Arabic script. And we do not have a consensus in the IETF yet on whether there should be an exclusion. We do, as we saw for Unicode 76.0, we really in the IETF want to stay with Unicode, otherwise, for implementers, it would be a mess if it is the case that we are a diversion from the Unicode standard.

But I cannot really say what the outcome will be. Thank you.

[APPLAUSE]

UNIDENTIFIED SPEAKER: I am happy that I can only write English and German. Okay. Any questions?

PATRIK FÄLTSTRÖM: They want lunch.

UNIDENTIFIED SPEAKER: Okay, we need to start at 2:00 sharp, because we have a compact program and I would like to get this done, more or less, on time.

UNIDENTIFIED SPEAKER: All right, if you can settle down please. Can we please settle down? Okay. Our next presentation is Kumar Ashutosh. I'm sorry that I misspell your name, because it's more complicated. He will talk a little bit about the Microsoft DNS management.

And I asked him in particular to put a bit of a focus on data mining, because that's one of the issues that some of us are grappling with at the moment.

KUMAR ASHUTOSH: Hello. Hi. I'm Kumar Ashutosh. You can all me Ashu or Kumar, as some people call me. And I'm going to talk to you about DNS traffic management. This is something that we have done new in DNS, with Microsoft vendors DNS server. And DNS data mining that is more from the [inaudible] because [inaudible] wanted me to talk about it.

So, here we go. So the idea of this presentation was to talk about Windows DNS Server and what are the capabilities that make it more towards Cloud ready. Now I know most of you, there is, ISC people, Bind people, and you have your own implementations, but Windows DNS server is also present out there, widely deployed in enterprises.

It has a fair presence in the DNS resolver space. Top 20 I think. It is standards compliant, at least we try to, and it's secure and scale able in terms of security. We support DNSSEC. [Inaudible] and scalable. As

you can see, most of our enterprise DNS servers, like [inaudible] Bing, etc. runs on Windows DNS servers, so that is scalable and really highly performed.

Now this is the first focus thing that I want to talk about. What are the needs of the DNS server in cloud? And the first and foremost need is a policy based traffic management, means how do you intelligently decide which query to be sent to which host? What are the response that you can send back to the client so that he can connect to the nearest or the best host?

There are many solutions available, like [inaudible], [vine] views, and there are other, well propriety implementations than those DNS server has come with its own policy based traffic management. Then the second thing is, because cloud is multi-tenant, you need audit and billing mechanism for the DNS service.

So if you are [inaudible] as a service, you can charge your customers, if required, so you need a proper audit train. The third thing is, you have so much data coming to you in cloud because it's operating in cloud scale, so many people out there, so you can mine that data. And when I say mining, you can find out the goal inside a big amount of [rock] and do something like [inaudible], I don't know whether it would succeed or not, but yes, you can mine the data.

And finally, the security and high availability, because if somebody puts his own DNS information on your infrastructure, he wants to be secured and highly available. Now the first thing, this is the first announcement of policy based traffic management, by the way, which is available in Windows 10 technical preview.

So DNS policy in Windows DNS server is a construct that allows DNS server to control the DNS query processing in order to achieve global traffic management, like distributing people from one geographical location to the closest data center. Application load balancing, which means I have multiple application, I want to distribute queries to each of them in a particular order, in a particular ration, depending upon the capabilities of those servers.

You can do that using the policy based traffic management. Then you can do intelligent DNS responses. For example, a person connecting to two, on IPv6 so get a [inaudible], or UDP and TCP based, some semantics, you can double up there. Applying tenant specific filters for black holing parental control, etc. And the major one, for a very long time, the split-brain deployment.

Now, how do we tackle our policies? How do we define policies? What is the anatomy of the policy? First thing that we have in our policies are the criteria. Criteria can be any combination of client subnet, means the IP address from where the client is coming, or the resolver's IP address.

Server interface IP means which server interface the query was received. If you have multiple interfaces running on the DNS server. FQDN, what kind of domain query. Internet protocol, v4 or v6, Transport protocol, UDP and TCP, time of day. At what time the query was received? Because in cloud elasticity, you will find application load [arrives?] with the time, because people in US will be querying for something at 6 to 9 in US.

But in Europe, the same application will not be loaded at the same time. So you could use DNS server to load balance your cloud data in the

cloud applications. And, query type, A [quad A?], SRV, TCP, MX, whatever query table, whatever you want to do.

Then, what happens…? And let me insist on this point, the criteria, when I say all of these things, value or variables, client subnet, interface, etc. you can mathematically combine them in any possible way. It is equal to this and this or this, not equal to this. You could do all of them.

So you could write a mathematical expression that will calculate into a criteria, and on basis on that criteria matching, you can take action to allow, deny, or ignore. Allow means pass the query, deny means block the query, send a failure response. Ignore means just sit back. Don't send anything. Let him rot.

And content is when you allow a policy to be executed, you respond with what answer? That is the intelligent response that you're going to give, and in what ration? If you have a load bland balance across different application data centers, you could load balance, A is to 2, 2 is to 1, is to 3, in whatever form for [inaudible].

Now what are the capabilities? I'll breeze through this. As you can guess, this is a very vast capability that we are introducing. And you can do traffic management, location of aware responses. You could do high availability by finding out which of your applications are failing over to where. And you can give the best answer available at that particular time to the client.

You can do the load balancing to distribute the queries across different application service. You could do time of day policies. You can do split brain DNS, because now if you have two interfaces on a single server,

one is external, one is internal. Internal is listening to the internal networks.

External is listening to the external networks. You could now define intelligent responses based on that. How and why, I will not linger on that, because I have to talk about DNS mine data also, but you can get in touch with me on how this can be done. I can just explain to you. And then black holing and filters.

Now the second part that I talked, was the multi-tenant, DNS audit trail. Now if you have multi-tenant, DNS… So many domains you are hosting, and you have many people who are administrating those domains, then you want to have a trail of what, who, and when? What changed? Like which zone changed? Which server changed? Which record was changed? Who changed? Whether it was done by a DC admin, or the data center admin, or whether it was done by a tenant admin.

And when it was changed. It was, it can be used for reporting, or the trails, diagnostics, because you have to give proof that you have done it correct. Now, earlier also, we had a basic DNS audit trail. Now we have introduced a separate [ETW] logs very fast, right in line, it's where you can get this in the form of XML data.

Now the final part of the presentation is the DNS data mine, where I will say, what we have available, and what I can, what you can use to do DNS data mining in particular. Now, as I said, so data mining is finding the most valuable information. So you do not care about all those data, but you want to find the most important data, and most valuable information out of that data.

How do you do that? This is a whole science in itself, but the basic DNS data mine, involves these four steps. First you have to collect data. Of course, you have not got to miss anything. Then there is data preparation. Like how do you clean the data? How do you wash the data? Then there is pattern discovery. You have several mechanisms to discover the patterns inside the collected data.

And then there is the action information. Now, first thing is the data collection. At present, Microsoft DNS Server provides you XML based logs for every transaction that is done on the DNS Server. Which means, you have a well formed structured document that can be consumed by different programs, and it has got IP address, FQDNs, which are the source IP, what kind of query, whatever comes in the DNS packet, it is available to you in a structured format.

And the best part is that it is done without decreasing the performance. So we have ensured that [inaudible] is the performance, when we do those, all those logging. And this logging can be scaled out on different infrastructure, and then you have to collect this data. The collection has to be real time, and it has to be with minimum performance impact, because for mining the data, you do not want to go and jeopardize your performance of the applications you are hosting.

And whatever kind of the data that you want to collect is, again, queries and responses, you want to collect the transfer request that have been received on your service. You want to collect the dynamic updates, or the DNS packets that have been received. And then you also have to collect the server state, like health indicators of the system.

Like CPU uses, disc uses, and all of those variables, and performance counters, which are very specific to DNS. All of these things are published by Microsoft in one way or the other. Now for the collection part, we are using a mechanism called ETW framework. And ETW framework is kind of a tiered mechanism, where there is a producer, which produces locks and gets back.

It goes into a buffered system, and anyone can write a [inaudible] to upload to a central server. Now, in case of data mining, the central server can be, what you call a relational database system, like [inaudible] server, which is multiple different databases collected to each other in an intelligent way, of course. Or it could be an overlap server, or the online analytical processing service.

Most famous of them can be SS, or SAP Net Weaver, or Microsoft's Microsoft Analytical Service. So you have that, then people have the [inaudible] application which do such kind of data collections. There is a Microsoft research effort called [dryad], which is very similar to the map produced by Google. A little more beyond that as well. And while using [dryad], you can collect data and do the data mining across multiple instances, multiple processes actually.

So, the second, the next part is the preparation of data. Now what do you prepare in the data, how to prepare the data. You first do the cleaning part, and in cleaning, after cleaning that, you have to find out which is the most important. How do you do data mining? Or, there is something shining, which is the anomaly, you find out that.

The first thing that you have to look for in the data mining for DNS is the anomaly. Where there was a peak, there was an amplification, there

was something that was going out of the way, that is the anomalous collection that you make. The second thing that you make, is classification. Now classification is a classic data mining way in which you collect and relate data, and how close they are, and how far they are when you are talking about a certain variable.

For example, if you have to, you have collected all this data, and with relevance to security part. Do you think all of the communications that are being done are secure? And they are totally not secure. So you can cluster the data that you have collected into secure data, and secure communication and non-secure communication. And somewhere between that, lies the place where there is a DNSSEC implementation without [inaudible] support, which is partially secure… Well, not so secure according to some, it's not compliant [in root].

So, that is one part. Then there is this clustering of data. The cluster is mostly used for user analysis, like user behavior and analysis, in which you cluster data which across the groups, and from a particular location in the world, how many queries will need for a particular domain. So which is the most popular domain in a particular area?

And these kind of data mining principle is called clustering of the data. Now, another thing that you have to take in mind is running over with the knowledge transfer, because you cannot keep the data forever. So what you have to do is to keep the information out, find out all of these things I told, keep it into information database, and then use it for the next iteration of the collection.

Now, the important part of this is after the collection and preparation is done, you have to do the pattern discovery. Pattern discovery has got

again, very established principles. And try to apply those principles to DNS in particular. One of the most prominent principles is the market basket analysis and [inaudible] analysis, in which you can find out what is the affinity of a particular client to lose something over and over again?

For example, if he has saw something or if he has done a Bing search on someone, what is the next thing he's going to click? Or for example, if he is buying a flower, what is the next thing that he is going to buy? He's going to buy a ring for his wedding, or something else?

So this is the [infinity] analysis in DNS, which can be used to predict the system. Like prediction of the user behavior. Now the second is the association rules, in which you actually find association between things which are not so obvious. A very good example of this, it is said that people who buy diapers in supermarket, also buy beer.

Now these things are totally not connected. How do you find that? So one of the examples, is for example, if some person is particularly looking for a domain name in some area, and after that, that is followed by an amplification attack, do you find a correlation between them? Can you associate them?

So this kind of association you have to establish. Between two seemingly unconnected things. And then there is this famous one, the sequential pattern mining in which one by one, if you find a sequence of events going on, you say that, okay, these are happening in a particular sequence, and this is the information that is extracted. And potentially the pattern discovery is done mostly from the domain name analysis,

like you find out which domain is getting hit hard, how many times, which part of the world is accessing this domain.

Then the amplification analysis. It's like you find out which kind of queries and where it's getting amplified. Are you getting amplification attacks somewhere? Then the user behavior analysis. Like what is the predictive of an user if he has gone to DNS for this query, what is the next query that he is going to make?

And that can be used for [advertisements] and other things. [Inaudible] analysis, like [rich client?] subnet is [inaudible] to doing more of these things more frequently than others. And the final one is the security analysis. It's like finding out where the DNSSEC is more prevalent, where the [dane?] is getting used. You people have talked about so many monitoring tools.

Mining data can be really wonderfully useful in that case. And the final, what is the actionable information? Actionable information in the cloud model are three. You have to do user behavior analytics, like what can you do? What is the next that the user is going to do? How do you scale? Do you need a particular dedicated reserve for a particular tenant or not?

Load model. Load modeling is the basic of cloud. Cloud is all about elasticity, right? It's the promise that whenever it goes, people come in and come out, you are given infinite resource promise to the tenants, and that no one ever feels that they have lack of resources. So if you're not being used in some way, you can [inaudible], you can use those resources for something else.

If you are able to find that the load model is going to increase next day, 7 to 10 PM, dedicate more DNS resources, or scale it down in a better way. And the final and obvious is [DDOS?] detection. You find out, as I said, if a particular domain, if a botnet is querying a particular domain at some part [inaudible], which is followed by totally unrelated amplification attack, how do you find them?

How do you correlate them? It is similar to buying a diaper and buying a beer. So, these are correlated in a way, but unless you do some association root processing, you cannot be able to do this. This is it from my side. I think I was, I did it in quick time.

If you have any questions on policies, or data mining, I'm all on. Particularly policies [inaudible] and questions, yeah.

UNIDENTIFIED SPEAKER: Okay. Thank you very much. Any questions? Thank you very much.

KUMAR ASHUTOSH: Thank you guys. Please connect with me.

[APPLAUSE]

UNIDENTIFIED SPEAKER: Okay. We have now two similar presentations, the CSIO, I think that's the Chief Security Information Officer, or the Chief Information Security Officer, I don't really know which… Either one works. First, the one

from Facebook, and then the room from Yahoo. Why don't you come up right here?

JOE SULLIVAN: Hopefully we won't give the same presentation.

UNIDENTIFIED SPEAKER: We'll see.

JOE SULLIVAN: Hi everyone. My name is Joe Sullivan. I am the Chief Security Officer at Facebook. And I appreciate the chance to talk to you today a little bit about how we think about security at Facebook and ICANN at the same time. Let me start by saying that, I guess, thank you. Facebook exists purely in the digital world, and would not exist without all of the work that all of you put into building out the commercial side of the Internet for companies like us.

And so, I think it's really important that we have a good dialogue together. That we have representatives from Facebook here, speaking for the people who use Facebook, and for our security team as well. Quickly what I'm going to touch on is a little about Facebook, a little bit about some of the things that keep me up at night, a little bit more about domain hijacking because it's something that we've experienced up close quite a bit.

Let me just jump right in and say, at Facebook what our mission is, to help make the world more open and connected. And what that means is we're trying to get everyone on the world on the Internet and use our

service to do that. And so we are a little bit unique in that we are a real name website, and so when people sign up for Facebook accounts, they use their real name. And we have had over a billion people do that.

Now, the screen shot I'm showing you right now is from this weekend. Our CEO was in India, he is now in Indonesia, and he's travelling right now because he's focused on something that we started last year called Internet dot org. And through Internet dot org, we're trying to bring connectivity to the four-fifths of people who are not on the Internet yet, and don't have that access.

And we've learned a lot, just in a few months of that project. Interestingly, a lot of the people who don't have Internet access, don't try to get Internet access because they don't appreciate the value of it. So even more than a lack of like technical ability to get the Internet is a lack of interest. And so, we're learning a lot, and through dialogue and partnerships with governments around the world, we're starting to make some good progress already.

The thing that probably is the most interesting part of my job, is the way that I see that people all over the world use Facebook to communicate, and how every conflict in the world, both sides are using Facebook to share their perspective, to promote their propaganda, and to attack the other side. It's a hard thing to be the platform to figure out which speech is okay and which speech is not, but when you put a security hat on, it's really great to just focus on making sure that people are speaking with their authentic voice, and that there is no, no one trying to compromise that.

We've had to get quite sophisticated over the years, because what we've seen literally in every kind of like dispute in the world, both sides are using different tactics, sometimes things we've never seen before, to try to get the other side's content off of Facebook, or to get their own content more promoted.

This I'm showing you is a great example of a page that our COO, Sheryl Sandberg, last week said that was her favorite page on Facebook, called my stealthy freedom. It's a page out of Iran. We had two pages out of Iran last week that got a lot of attention. One on Instagram, which was, Rich Kids of Tehran, which was somewhat humorous because it was rich kids putting up pictures of their stuff on Instagram.

But I think this is a lot more important page, my stealthy freedom, where real women in Iran were taking pictures of themselves without head scarves. And so, like there are people who do not want this content on Facebook or on the Internet. And what we see around the world is different types of attacks.

A really interesting one recently was in Southeast Asia, there was a kind of political discourse going on, and one side tried to use the reporting mechanisms to take the other side's content off the site, and they finally figured out that the one place where we really, really, really aggressive in automatically taking things down, was any time you report someone as under the age of 13. Because the law in the United States is incredibly strict about companies retaining data of someone under the age of 13 without their parent's consent.

So our reporting mechanism was hair-trigger on that one specific issue. And so all of the sudden, everyone on one side of a political issue in this

country was reported as under the age of 13, and then we had to learn from that experience. And so this is the types of things we're seeing all of the time.

Sometimes when there are major elections, every domestic or regional smaller websites get denial of service off the Internet. And we'll be the only site that people can communicate on. And, you know, we have certainly seen our share of denial of service attacks in those contexts, and we've also found that we're generally able to withstand those because, one site's denial of service attack is our world cup.

So, I think what you'll find when you talk to people like Alex or me, is that our security teams, we're really blessed. We get lots of resources to spend on security, and so that's why we're able to spend time coming to talk to you. We're able to do get involved in open source projects. We're able to get involved in WC3, we're doing blog posts talking about the work that we're doing, we're putting out lots of technical papers.

Just I'll mention, two areas where we provide a lot of information, is we have a Facebook page called protect the graph, where we, in the last year we've published details about two open source projects that we've released for security purposes and things like that.

When I was coming here, I asked the team, "What are the things that you would like me to raise?" And the number one thing was, we do a lot of investigations into abuse. Most of the time we're not able, like everyone else, we're not able to get law enforcement involved, so we have to investigate things and learn what we can for ourselves to try and protect the people using our services.

And our investigations run into dead ends all of the time because of bad WHOIS data. There is definitely some concern about name collisions coming. I'm not sure how real it is, I guess we'll find out. And then there is this general unease about perceptions of entrenched interests and things like that, that we always have to deal with.

But like the thing I wanted to talk about for five minutes today was domain hijacking. We have an interesting experience back in… Well, I should say before I came to Facebook, I worked at Ebay and PayPal and I moved to Facebook in 2008. So take you back to 2008, we were dealing with phishing. Phishing was a big thing back then for PayPal in particular.

And so, on this particular occasion, PayPal reported a phishing URL to a registrar team, the same registrar team that was hosting PayPal. And they got confused, and they took down PayPal instead of the phishing site. [Laughter] So, that's not something that you want to have happen.

UNIDENTIFIED SPEAKER: Oops.

JOE SULLIVAN: And to make matters worse, it was Good Friday in the afternoon, and it was really hard to get back in touch with someone and get things straightened out. So, actually Susan [inaudible] who is here representing Facebook, worked really closely with our registrars to get setup what's now called a registry lock. And that proved really valuable,

and that's something like we're a huge fan of, and I'll explain a little bit more while we're such huge fans of registry locks.

I looked into it, and I was glad to find out that the security and advisory committee actually solved domain hijacking in 2009. I don't know if anybody has read that paper, but they, you know, they were looking at a number of major attacks against large sites. And I think it's the same kind of issues you'd expect to find everywhere on the Internet around security. Nobody is sure whose responsibility it is, most people aren't crossing their T's and doing all the little things.

It's really hard to tell which services have better security in terms of when you're signing up, and most people assume that they want to get the cheapest version rather than the most secure version. And then, in 2010, the problem didn't go away. The was the headline. The Iranian Cyber Army, took over Baidu, which is one of the largest search engines in the world, and put this nice dark page up.

And you'll see this is a theme. So this is the Iranian Cyber Army. They also, I know of one other major company, I can't say their name because they wouldn't appreciate it, but at the same time as this, their registrar received a fax letter asking for a change email address, and they actually changed it, and then that site would have been compromised. At the same time, Facebook dot com was attacked, and we know that because we had a registry lock in place, Facebook did not end up with this headline.

So, jump now forward to 2014. This is an internal Ebay and PayPal email that was published on the Internet. Allegedly, the Syrian Electronic Army, spear fished an Ebay or PayPal employee, then used

that employee's email address to reset the registrar password. So now you have outsiders with control over an internal email that happens to be on a whole bunch of internal threads, dealing with the incident because their site now looks like this, PayPal dot co dot UK.

And despite the coloring, this is not the same electronic army, this is, that was the Iranian Electronic Army, this was the Syrian Electronic Army, taking over PayPal's site. And then you saw, they posted that they had access to [marc?] monitor, because they had reset the password of that PayPal employee. So this was in 2014. And then, after PayPal cleaned up, you saw the Syrian Electronic Army was directly going after [marc] monitor, and probably other registrars as well.

You can see here, this is where they compromised a [marc] monitor email account and they're testing it. We got this from somebody who was helpful. And then you see what happened to the [marc] monitor report, it had to go off line. And while they were attacking [marc] monitor, they went after Facebook. This was actually posted on Facebook by the alleged Syrian Electronic Army.

And as you can see there, they're posting the WHOIS lookup info for Facebook dot com. And it shows that they were able to change, once they got access to [marc] monitor, they were able to change our street to Syria, our city to Damascus, and our state province to Syria. But where they were not able to change was where it said domain status. And that's, again, because we had the registry lock in place.

So, the Syrian Electronic Army there was reporting to a compromised Facebook, and if we didn't have a registry lock in place, then we would have been in trouble. And then you can kind of flash forward to today.

This was another dark screen for Google Indonesia, this time it's [mad leaks], reportedly associated with the Pakistani Cyber Army, also choosing the dark screen. At least they were courteous enough to leave a search bar on the page so that people could actually do a Google search from their hijacked URL.

So, you know, the short version is, domain hijacking is still going on. If the biggest and best brands in the world, that are investing the most in security, are still under attack like this, with the best solutions we have in place, then we probably have more work to do. If we don't work harder, the neighborhood is going to end up looking like this. And you know, we're all here because we want the neighborhood to look like this.

So, you know, it's, there are a few things that need to happen, and they're all tied to security, security awareness, collaboration, best practices. But like we really need to get together on this stuff and do a lot better, because I'll tell you, what I see with the little people who have accounts on Facebook, who are trying to have a voice, when their accounts are attacked, it doesn't end up in the headlines when they get compromised, and it's the same in this context as well.

They're not ending up in the headlines, but they're losing their identity online. And so we've got to make sure that we get high quality security standards in place for domains across the board.

UNIDENTIFIED SPEAKER: Okay. Thank you very much. Any questions? Wow. Then at least give him a hand. [Applause] And now we'll have the perspective from Yahoo's side.

ALEX STAMOS: Hey, folks that are standing, there are a bunch of seats up here. If you want to sit down, let's give you guys a chance to…

UNIDENTIFIED SPEAKER: That's actually quite correct. People who want to stand, can remain standing, but we still have a number of seats here.

ALEX STAMOS: Hey Christina, is my file on here? Okay. Oh here it is. Let's see how do I do this? Oh.

Hey everybody. My name is Alex Stamos. I'm currently the CISO at Yahoo. And happy to be here. I've met a lot of you folks in my previous duties at what is now NCC Group Domain Services, and I moved over to Yahoo in March of this year. And I've been very happy to join the ranks of Joe and the other very welcoming CISOs who let me into their group of people who don't sleep too much.

Anyway, I was just going to talk a little bit about, less relevant directly to the domain name industry, maybe more of an IETFE kind of discussion, but I thought there was a lot of people here who would be interested to hear about what we are doing in the industry as a response to revelations of widespread Internet surveillance, and the kinds of things we'd like to see technologically to come out of groups like this.

So what are, wow, this is really bad.

So what kind of Internet surveillance revelations are we responding to? Obviously government tapping of public networks, pretty much every government we know does some amount of this. And it has turned out that a number of industrialized nations very widespread ability to record unencrypted information on major points and presence around the Internet and index the data for later use.

Government tapping of private networks. There have been a number of revelations of the very expensive backbones, that companies like ours pay for, have been tapped by government folks, who then write slide decks complaining about how much spam gets moved between our data centers when we do backups of our entire mail spools, which is obviously problematic.

Adversaries tapping local networks. Obviously not all Internet surveillance is widespread. A lot of people in the world do not have a secure local connection to the Internet, and we shouldn't discount the ability of less advanced attackers to do kind of less widespread attacks. Adversaries getting access to user accounts. So a constant problem in the consumer Internet space is, it is very difficult to establish and maintain a secure connection between an user's account and their identity, their actual [inaudible] space identity.

And as a result, the overall user account lifecycle is very difficult to maintain, and users have their accounts taken over on a regular basis. For our own purposes, we find that whenever there is a major dump of credentials to the Internet, that 10 to 15% of them are accurate Yahoo user names and passwords. And so, you know, if we have a credential

dump of 120 million accounts, that means like we've got 17, 18, 19 million accounts are Yahoo accounts, that we now know are potentially compromised, and we have to reach out and somehow reestablish trust between us and our users.

Which is very difficult in a situation like ours, where we don't require real names, we don't require people to have a social graph, we don't have some of the benefits Facebook and other social networks have to reestablish that. And then we're also responding to lots of evidence of certificate authorities behaving badly, usually on behalf of governments. Earlier this year, an incident that got almost no play in the press, which I thought was amazing, the Indian CCA, the Indian government's certificate authority, which I believe is operated by university on their behalf, got caught creating certificates for Google dot com, Facebook dot com, and Yahoo dot com, which they then played off as an accident.

Amazingly, somebody tripped, fell on their keyboard, and created, you know, 20 subject alternate names for exactly the things that you would want if you were trying to man in the middle and steal the passwords of their users when they're logged into Yahoo. So obviously, these kinds of accidents happen, and we need to have protections in there so that when these accidents happen users are not impacted.

Obviously, other governments do it, they just do it in a way that they're better at not getting caught. So what have we done in response to these things? So we've moved to encrypt all of the data on our frontend, using transport encryption. Generally TLS 1.2, perfect forward secrecy, which is extremely important. We don't see it everywhere, and

it's very important because it turns out that the thing you need to decrypt a TLS session without PFS, is the private key for a certificate.

And that's the thing that a web scale company has to push out and make available to tens of thousands of frontend servers. And so as a result, this thing that is super-secret, has to be on tens of thousands of machines, that are by definition are on the Internet, is a difficult circle to square. And so, putting PFS on means, while it's not great that you get your key compromised, at least it does not re-enable widespread Internet surveillance, although it does enable obviously small scale man in the middle attacks.

AES-GCM, RSA 2048, and for the most part, we're now admitting HSTS headers for most of our sites. Things that we need to be working on. We're working on pre-loading pins into the browsers. Like I said, we cannot trust the browser manufacturers or operating manufacturers, we cannot trust their list of certificate authorities. There is something like 300 organizations listed in Windows as certificate authorities that are trusted, over a dozen of those are directly tied to governments, including our own, and we can't trust any of those.

And so we need to a way to limit the impact that a [inaudible] authority can have. Right now, the best opportunity for us is to preload pins into Chrome and Firefox. Unfortunately, Internet Explorer and Safari do not support that, and there is very little we can do for those users. ECDSA certificates. So who here has heard of the [shaw 1, shaw 2] 56 transition?

Yeah, so there is this crazy thing going on right now, where most of the certificates on the Internet are going to have to get changed in the next

year, because the major browsers are starting to disallow the use of [shaw 1] in the signing algorithm. The downside is, [shaw 1] is not supported by a huge chunk of the people who use the Internet, people who run Windows XP, people who run Android pre-version 4.3, I think, people are on lots and lots of phones and computers out there, can't handle [shaw] 256 certificates.

And so what we're doing, I think most big folks are doing, is we're building a new stack on our frontend that has the ability to intelligently route certificates for people to come in. And so if you come in with a new browser, you're going to get an ECDSA certificate on the [inaudible] P 256 curve, as well as [shaw] 256. If you come in with an old browser, you'll get a RSA 2048 [shaw] 1 certificate.

So we'll provide great security to the new browser, not as great security but acceptable security and not cut off the old ones. We use a software called Apache Traffic Server, which used to be called Yahoo Traffic Server. So if you use ATS, then you'll be able to get those patches right out of the public repository. We're working with our CA vendor to do certificate transparency, which is a mechanism that allows us, and the rest of the world, to verify that our CA is not acting poorly. Unfortunately, certificate transparency is highly controversial among the certificate authority members of the CA browser forum.

And so it has been very difficult for us to find a CA that is both used widespread enough, as well as willing to do certificate transparency by default for their master root cert. So we might have to get our own intermediate certificate authority so that we can do this. And then

we're looking to implement ChaCha 20 and Poly 1305, we're turned on their cyber suites.

We've had to do a lot backbone encryptions. This is a very simplified map of our Internet backbone. Each of those little squares is a datacenter. So these are datacenters that we own, these are not datacenters that other people that we have pops in, which we also have, that's a different map that's so complicated that doesn't show up on a slide. It just looks like spaghetti.

You know, after the revelations that folks were tapping our backbone links, we put in a crash program. And as of March this year, all of the traffic moving over our backbones outside of Yahoo physical plant is encrypted. This is done through two ways. One is through the obvious, go through money at the problem solution, which is you buy very large boxes from network providers that do IP stack or other network layer encryption at wire speed.

This turns out to not be that easy. There are not a lot of companies that make boxes that can encrypt traffic at upwards of 10 gigabytes per second. The smallest of these links is 10 gigabytes per second, and then they get much larger from there. And unfortunately those boxes have the tendency to break, and when they break, you lose connectivity between two data centers. And at a company of our scale, if you do that for multiple days, you end up with a data coherency issue, measured in petabytes between two different datacenters.

And so what we're pushing to do, and I think most people are going to have to do is do TLS point to point between all of our servers. And then it goes unencrypted over the backbone, but it's encrypted. And so

we're already in a situation where we do policy based routing. So we have a bunch of stuff that's encrypted and verified using TLS between servers, and that goes around the IP set boxes.

I think forever, people are going to have to have IP set boxes, because there are some things that you just can't tunnel over TLS. But for the most part, you know, both data transfers are going to have to be point to point for companies of our size. We're working on the last self-service security for our groups. So we have over 80 products that have to be internationalized into 60 countries that we support.

And for our web… We do continuous integration, continuous deployments, so there are groups at Yahoo that are doing multiple pushes to the web per day, and then weekly mobile pushes to the app stores. Obviously that makes it impossible for us to keep up with it. And so, you know, a good, something for us to do and a good thing for companies our size to do, is to solve those problems once with standardized libraries, and then push that out to everybody.

The great thing about standardized libraries, is when people don't use them, it shows up really easily in [inaudible] scanning. One of the great disappointments of the entire security community, is that, you know, stack code analysis still doesn't work, even though we're 20 years into it. But one thing that does do well is tell you if somebody is talking, doing TLS without using the TLS library. We're talking to a database without using the proper database library.

This is supposed to be a video. Oh my God, did they create multiple slides, okay good. This is supposed to be a video, which isn't going to show because of how we're doing Adobe Connect, so that's fine. One

of the other things we have to do from an encryption… So obviously, transport encryption is great between users and us, and between our mail servers and our mail servers and other mail servers.

But we also want to provide the opportunity for users to do better than that. And so, earlier this year, we announced that we are working on end to end encryption for mail that will be rolling out next year for all of our users, as an option. The way we're doing this right now, so our browser plug in that watches you type an email out in Yahoo Mail, which is the Yahoo Mail web client, sees whether or not the people you're talking to have keys registered.

If they do, it says, "Hey, do you want to encrypt this?" You say, "Yes," and it silently replaces the window you're typing into with an iframe that is controlled securely by that plug-in, and now you type into a window that only exists on your computer, it encrypts on your computer and then it moves through our network completely encrypted.

So we have no option or way to attack that. We never have your private keys, we never have your plain text. And we're working on this with Google, with our goal of having completely compatibility between Gmail and Yahoo Mail, so you can type in a Gmail account, and it will turn green and say, "This person has encryption," and vice versa. If you're on Google and you type a Yahoo Mail, it will detect that.

This requires, obviously, going way beyond what the current RFCs allow, and so it's opened up a lot of opportunities for standardizations, so that we can do the same kind of thing with smaller providers than just, you know, the top two or three. So [inaudible]. One of the kinds of things

we need to do as an industry, we need to do an IETF to support preventing Internet surveillance on a widespread basis.

So we need for future crypto systems to always have the ability to have non-list cyber sweep options, in the full stack. So a problem that international companies like ours are going to be facing, is that we are looking down the barrel at likely requirements by the EU, Brazil, Russia, a number of other countries to not use cyphers that came from the US government.

Thanks to revelations that at least one missed standard was backdoor by NSA. And so we'd like to get ahead of the government mandates, I would hate to be required to use a different cypher suite that we don't trust. That comes from a government. I would rather, as an industry, we come up with these cypher suites, and establish trust in them, to hold off some of this. So obviously, ChaCha 20, Poly 1305 are pretty popular, as is Curve 25519. Right now, we have no option for certificates outside of either RSA, which has a number of problems in that, RSA keys are now so large they don't even fit in one Ethernet frame.

Which gets really problematic when you're talking about people on [bubble?] networks, right, having humongous packets going back, you know, packets to do handshakes. And the other, so we want to do ES DSA, which we have no choice but the missed P curves, which backdoor or not, no longer conform to some of the things we know are requirements for having safe elliptic curve cryptography.

And so we're really hoping that future standards give us the options that we can support these things right now, and we don't have to wait

for the EU or other countries to act. We need a standard for pinning, encrypted SMTP between mail servers. So we do this manually between us and the other major servers, by us calling them and asking, "Is this the correct key for you?"

And then putting manually into our software require a key when you talk to Google, or we talk to Hotmail. That's not very scalable of, you know, mail companies knowing each other's cell phone numbers. And so, there definitely needs to be a standard way either in band SMTP, or via some other advertisement, publically that Yahoo will only ever, will always support from here on out, encrypted SMTP, and here is the certificate authority that is allowed to sign our certificates.

Until that's done, then widespread interception is going to continue to happen, and it's happening right now. There is a number of countries that are intercepting SMTP connections, inbound to any ISP in country, and stripping stark TLS. And there is very little we can do, from our side, without a technical standard that we can use to communicate to those folks to prevent that from happening.

I think, anybody here work on the HTP biz, anybody here work on HTP 2.0? Yeah. So you guys are making a huge mistake if you're in that group by not requiring optimistic encryption. I think there is all of these philosophical arguments, that this should be in TCP, yadda yadda. The truth is, the only way this is going to happen, is it's going to happen in HTP, and by now doing it there, you're basically saying that the Internet doesn't deserve privacy from spying eyes.

That's the practical impact. I don't care what you're philosophical arguments are, they're all kind of crap from what I've seen. And so, it's

a huge mistake. I'm really glad that Chrome and Firefox have announced that they're going to require encryption for HTP 2.0. We're moving very aggressively, obviously, to support it ourselves. But overall, I mean, we need to put down the philosophical OSI layer arguments, of whether something should be in the transport layer or the application layer, and we need to do what's right for users.

So hopefully, HTP 2.0 is not done yet, hopefully we can make a last minute change to make options to encryption a required part of that. And I think we need as an industry to think of a replacement for open PGP, as a standard. The open PGP, PGP is done its duty, but we know of a lot more ways, both a lot more mechanisms that people want to have encrypted messaging over, than just email, as well as a lot of ways to keep it safe.

And as we sit down, and we're working on our end to end encryption, we're using PGP right now, because it's pretty much all we have got from a standards perspective, but it prevents us from doing a lot of cool things. For example, we live in a world where you can have searchable encryption, where messages are encrypted, and they can be searched on the server without the server having the key to decrypt that message.

And that's something that is mathematically possible, but there is really no way to retrofit that to an old standard like open PGP. There is no way to fit an open PGP message into 140 bytes, which means that transport layer over things like SMS is impossible. And we don't have things like [inaudible] secrecy [inaudible] the other kinds of modern cryptic mechanisms is pretty much impossible to add to the standard.

I would love to see a new RFC standard here, that's kind of an umbrella standard, that has all of the different options, and then you have different profiles were to use. You have a male profile, instant messaging profile, maybe a mobile profile, where some of these things are used and some aren't. And that we can have one standard way of doing it, because right now, it looks like we're going to be fracturing into multiple different implementation standards that are just standardized in the implementation and they are proprietary for a company.

Another thing that I just wanted to bring up in this world, well I'm not against DNSSEC. There does seem to be a focus on using DNSSEC as a bootstrapping technology for anti-surveillance technologies, that is not very helpful. There are some fundamental problems that DNSSEC, that I don't have to repeat at length here, but the idea of centralized keys, is a complete non-starter.

In a world where the companies that control those keys are domiciled in countries that are known to put pressure on business to give up private keys or to do bad things, I know a lot of nice people at Verisign, I have no problem with them, but I can't trust Verisign as a company. It's just not possible. Verisign is a US government contractor. The US government is their number one source of revenue.

There is absolutely no way I can trust Verisign not to do something bad with Yahoo dot com sign zone. And so, as a result, I think we need to have anti-surveillance technologies have to be, by nature, decentralized, or they have to be centralized with a decentralized mechanism to keep people honest. Right?

So you know the fact that DNSSEC has no equivalent certificate transparency built into it, while you know, there is theoretical ways you can do that, until those things happen, then it's totally a non-starter to use DNSSEC for anti-surveillance technologies. So I would like to see more tofu trust on [inaudible] technologies, more optimistic technologies, as well as technologies that take into account the fact that the Internet is naturally asymmetric.

We don't have to, you know, when you talk about things like, you know, web services and stuff, a handful of big companies are a huge chunk of the Internet, and so the asymmetry of very large companies operating services for hundreds of millions of people that, even if the technology doesn't work for every single little webhost, that doesn't mean it's a bad thing.

Cool. And that's kind of my spiel. So, any questions?

UNIDENTIFIED SPEAKER: Thank you very much. Questions on that side.

ALEX STAMOS: Oh, a couple, okay.

ANDRE: Hi, this is Andre [inaudible] from [inaudible] and former [inaudible] co-chair. I know you don't like DNSSEC, but for SMTP pinning, we have it, it's called [dane]. And, well… You should have at least mentioned that.

ALEX STAMOS: No, so my point is, there are all kinds of cool stuff build on DNSSEC. DNSSEC is not, one, it's not end to end. Two, does not have decentralized trust models. Three, is not deployed widely enough to be used, and therefore… My kind of point is kind of like this focus on DNSSEC as an enabling technology, is plugging up the whole pipeline, right? Because it's kind of like, first assumed DNSSEC works, then we can have these things.

And so I'm glad [dane] is happening, but honestly, you guys are building this edifice of security technologies on DNSSEC when a fraction of people on the planet get DNSSEC, will actually have resolvers that verify DNSSEC for them in a secure way. Plus you can't, we can't have resolvers… We have to have end to end technologies, we can't have resolvers doing it for us, because the resolvers are controlled by ISPs, and ISPs are controlled by governments, right?

Yeah, so I think it's great that you guys have [dane], but that's not going to do it, so I need something else. I prefer something that is in band to SMTP, that's just a header that says, "Hey, from now on, kind of like HSTS or HPKP." From now on, please always use start TLS. Because those kind of trust on first use things, while theoretically not secure in some kind of fantasy world, in practice are secure enough to prevent the kind of things that we're worried about.

WES: Wes [inaudible]. And I'm going to speak more about DNSSEC, so I'll be giving a talk on DNSSEC and SMTP on Wednesday, please do come to it. And I would love to talk with you offline about the issues. And certainly, your concerns may be justified if you don't trust certain points in the

tree, and you're anchored below them. The standard answer is don't anchor yourself below something that you don't trust.

Because there are always going to be a problem anyway, right? They can redirect you anyway. The important thing to note about SMTP though, is that there nothing you can do with the start TLS layer that doesn't prevent somebody else from going later on, I don't support start TLS as a man in the middle. And if you do the pinning, and the leap of faith like you're talking to, that only assures that you're still talking with the same man in the middle as you originally started talking to.

It's not really a security solution.

ALEX STAMOS: See, so that's the thinking that gets us in the wrong place. Sure, there is a hypothetical, but the first time we talk to an ISP, we're going to get man in the middle, and that's true. The truth is that for, you know, like the connections between us and Gmail and us and Hotmail, are like the busiest SMTP connections in the world, and between us and most ISPs are extraordinarily busy, so that's the kind of situation in which we would notice that and break, right?

And I would rather have something that right now solves my real problem, than this theoretical problem that solves it perfectly, that will be deployed in 2027. Right?

WES: I can give you a list on Wednesday of a whole bunch of major deployments that have happened on the [inaudible] front, so it actually

is happening. It's much easier as the SMTP server, than it is, yeah, at the ISP all the way to the desktop, that's a much harder problem than actually running a local validating resolver on your mail server directly, which is what the recommended approach is by a lot of operating systems.

But, forgetting that for the time being, there is no way in transport to do it securely. You have to have an external bit that says whether or not this thing is going to be able to do security. Yes, you can pin certain things, you can pin Google, you can pin a few, you know, and get 90% of the world, but you're still subject to the other 10% being a loss. And I hope, I hope, hope, hope that you have some agreement with people like Yahoo, people like Hotmail and Google, where when they change their certificate, you're going to know about it in advance, because otherwise you're going to break some day because they just had to renew their certificate and you didn't know it was coming.

That sort of manual configuration only works at a very small layer…

ALEX STAMOS: Right, I don't want manual configuration. I'm just saying trust on first use is a totally acceptable security model in 99% of situations. And I'd rather have trust on first use today, then wait for [dane] to work. Now maybe SMTP is a good example because we have the ability, we don't have to wait for end to end DNSSEC, so it's one of the few situations in which DNSSEC and [dane] are actually useful.

But I'm actually a bigger supporter of the start TLS everywhere project the EFF. Which is a pretty simple, you check into a get repository of

what your config is, it gets signed, and then everybody sees this one file, and you check to see if somebody is lying about you. And so, maybe a solution where we use [dane] for the smaller folks, and something like that for the large folks, is where we'll end up.

JOHN PETERSON: Hi, real quick. John Peterson.

UNIDENTIFIED SPEAKER: One more question.

JOHN PETERSON: One more question, thanks so much. Please come to the ITEF. It's a great presentation, I think people in the ITEF will really want to hear from operators like yourself who are in this position. I can tell you though, the people that say Mozilla, who build Firefox, who are deciding how TLS and things like that are going to work, are the people that are arguing about [tofu] at the transport layer and things like that in those working groups.

And they are building this. Like this is stuff that is not 2027 stuff. This is stuff that they think is going to be, you know, in Firefox and Chrome very soon. And so, I mean, this input would be great, but this is the wrong place to do that, if you want to influence the people who are actually going to make this happen or not.

ALEX STAMOS: Okay.

JOHN PETERSON: No, I'm saying seriously, please come to…

[CROSSTALK]

ALEX STAMOS: …fine…

JOHN PETERSON: Please come to the ITEF.

ALEX STAMOS: Okay, fine.

UNIDENTIFIED SPEAKER: Okay. Our next presentation is already lined up. Alex will talk about DNSSEC.

ALEX ROUSSKOV: My name is Alex Rousskov. I work for the Measurement Factory. And I'm going to talk about DNS Rex, which is DNS performance test tool. If you have small kids, or remember the movie *Jurassic Park*, then you know that tyrannosaurus rex was one of the most aggressive animals out there. But I don't want you to think about DNS Rex as this vicious animal that destroys your DNS servers.

A better way of thinking about DNS Rex is a vicious animal that you can control, and configure, and tell it exactly what to do, and then receive a detailed, nice statistics report, after it destroyed your web servers. Our DNS servers.

Okay. So, a brief overview of what has happened in DNS Rex life. As I said, it's a performance test tool for DNS servers. In 2009, we were approached by a customer that needed help with performance testing, and they wanted to test performance of a powerful DNS server, with a few caveats. First of all, they wanted to test a resolver mode, and they wanted to test how effective certain advanced cache [poison] measures were in that server.

And so that's what we created, that's the tool we created. And in about a year, since they started with a project, we rolled the software, we ran the tests, and declared the mission accomplished. This software was publically released under an open source license, and pretty much nothing else happened since them, which is one of the reasons why I am here. If this situation doesn't change, the software will die a slow death and disappear.

So the first question that many probably ask is, why on earth we created another DNS benchmark? And I can honestly say that we resisted that temptation for as long as we could, and the initial plan was to use existing software to run all of those tests. Unfortunately, we quickly found out that existing software focuses on authoritative servers, and not resolvers. We also needed to test cache [poisoning] defenses, and although there were some tools available for that, in general, they were slow, unreliable, or shady, or all of the above.

There was also another, somewhat strange problem. Whenever we talked to this customer or others that work on DNS, there were certain distrust and dislike among developers of DNS servers, for the tools that were already available to them. I cannot remember a single instance where a person I would talk to would say, "Oh, we love DNS [curve]," or another tool. That never happened, and usually the emotions were quite extreme on the opposite end.

And first, it was surprising to me, but as we collected more data, and this example is one of that, it became clear that there are some facts behind those feelings. So here is an example of us testing a resolver, using two tools, tool A and tool B. The first tool claims that the maximum throughput of that DNS resolver is 22,000 queries per second, and at that point, it was losing 24% of queries. Tool B conclusion for the same resolver is that it can sustain a throughput of 120 queries per second, while not losing any queries, answering all of them.

So when an engineer looks at this, obviously at least one of them, at least one of these conclusions is wrong, maybe both. And that doesn't develop trust. So, all those factors, plus the fact that we had a lot of experience creating HTTP performance tools that also deal with high loads, caching, and related issues. We sort of… We could foresee problems with using the tools that I have available out there.

And we resisted as much as we could, but at the end, we had to create something that worked. So we gave up. During that analysis, it was interesting to see that there wasn't that much progress in DNS testing tools, and now a few years since then, it becomes even more evident that the whole area is kind of stuck. I don't know why that has

happened, but I can speculate about a few factors that have contributed.

So it looks like the easy problems of testing a DNS server have been solved. You get, out of the best tools available, you get something like this. You get a bunch of HTTP queries that are being sent at an increasing rate, whenever you find enough errors, you build or continue sending at the same rate, and say, "That rate is the maximum throughput for that particular DNS server." And so, that's it.

We declare victory. The problem is solved, whenever you want to test performance quickly, you use that tool, it gives you a number, and you are supposed to be happy.

Unfortunately, even that sort of simplistic approach, doesn't really help these days. So since roughly 2007, there have been no performance improvements in CPU speeds, if you look at a single quark. So, all those tools that were using single quarks couldn't really keep up with the progress of the DNS servers that were obviously threaded by them. And if you realize that, yeah, now you have to thread your tool, then solving all of these difficult problems that those simplistic tools haven't solved yet, becomes exponentially more difficult because now, not only you have to deal with threads, you have to deal with complex problems while threading your software.

So the remaining issues are hard to solve. And maybe that's why there hasn't been significant progress, or any progress. One speculation is that the suppliers of popular tools, have to focus on other things now, and they have to focus on their own products and their future, and they

just don't have the cycles to continue to develop tools that everybody else can use.

And perhaps there is insufficient demand as well, though from what I've heard so far, I kind of doubt that.

So if we have to move forward, I think it would be useful to understand what we want to move forward towards. And these are the kinds of questions that we have to answer when we started developing DNS Rex.

So one of the key factors, in my opinion, is persistence, which in this case, I define as sustaining the configured load for more than a few minutes, or in some cases, seconds. So if you look at testing instructions that come with some of the popular software, you can find something like this. You know, we migrate from 3 million records in a query file, to 10 million records in a query file, because 3 million is just not enough anymore.

And that was in 2012, so I guess, when the tool is updated, we'll have 100 million record files, and that just becomes unmanageable at a certain point. Moreover, when you think about current query rates, 10 million records do not really give you much. You know, 100 K queries per second, that's just 100 seconds of a test. And if you look at a recent 2014 DDOS threat report, the longest single attack that they measured was 9 days and 11 hours, so obviously 100 seconds, 9 days doesn't really match.

So you need something better than just a trace replay. Another issue is scalability. And here it's sort of divided in two areas, one is scalability on a single box, and there it's kind of simple when you test a DNS

resolver, or DNS server, that runs on a single box, you get a similar box, and you make sure that your software is faster than the DNS server software, and then you're in business.

Unfortunately, since some of the most advanced DNS servers are run on custom, or very expensive hardware, it becomes difficult to test using the same hardware in your lab. So then you move onto a different configuration, which might be called [swarm mobility?] or swarming, where you have a bunch of cheap boxes producing that load. Needless to say, in this case, if you just have an engineer that starts 10 processes, on 10 boxes, controlling that test and aggregating results after the test, becomes a challenge.

So what's the worst case scenario? In this case, well again, from the same report, they measured 23 million queries per second as the worst DDOS rate. I don't know if that's the maximum still, but that's a pretty high number. I don't think a good benchmark that uses off the shelf hardware and no specific hardware tricks can give you that rate from a single box, but maybe if you do one million, maybe half a million per box, then you can scale to levels that approach this kind of attack.

Cache awareness. This factor is specific to resolvers that cache. None of the tools that are popular right now are cache aware. It would be nice if you could configure the hit ration that's present in the traffic, so that you can test your resolver at 100%, at any percentage in between , and maybe actually vary the percentage through time. If you are using traces, then with a short trace you would get 100% hit ratio, which is not realistic. If you somehow get an infinitely long trace, that is still not a solution because it's a fixed hit ration percentage with that trace, and

to change that, you would have to change the trace, which is usually difficult and time consuming.

This is somewhat an unusual property for a test tool, for a performance test tool, it needs to be able to slow. So when you test something like a DNS resolver, you want to delay answers on the other side, on the authoritative server side, so that the resolver has more concurrent transactions to deal with. And that's difficult with the current tools. It would also be nice to simulate things like packet drops and [inaudible] domains, better for all of those errors and what not.

And independent, there are several layers here for a resolver test, ideally you don't really want to use bind or your own proprietary authoritative server, you want something that is independent from what you are testing. And unfortunately, that's not the case today. It's hard to avoid that, and there are several reasons why you don't want that. I'll skip them to finish sooner.

Another layer here is that you may want to avoid resolver libraries, again because you may be afraid of certain hidden bias in your tool, if it is using the same software that the DNS server that you are testing is using. And the most extreme range, you may say that the benchmark should not be developed by the people who are trying to sell you something, because they have a bias.

Although that's probably an extreme point of view. An obvious range of various protocol features: IPv6, TCP, DNSSEC, which is especially hard because it deals with encryption, and you cannot do that fast while you're testing, you have to pre-sign zones and stuff like that, which takes a lot of time.

And these are views, we all want that, although it's not really clear what views are in a sense, when you are talking about performance benchmark. Configuration files are nice. You can document what you are configuring. You can share them easily. Perhaps more important point is awareness of the test environment, and sort of being able to understand that something is going around and warn the user about that.

And I'll skip other important things. So, coming back to DNS Rex, from marketing point of view, we've accomplished almost everything. You see a greenlight across the board except for IPv6 and TCP, which were out of scope for our project. The reality is a little bit different. Most of those things need more work, need more development cycles, but still, it is pretty reliable, it can scale at least on one box, it is threaded, it has lots of configuration options that I already talked about.

It can configure hit ratio, it can assimilate slow DNS servers. It is independent. There are some DNSSEC nodes and so forth.

So, that's where DNS Rex is today. What we do not know is whether we should continue working on it, or just let it die. If we are to continue working on it, we need to assemble a group of people, companies, who want to push it forward and try to create another popular DNS benchmark that everybody can use.

If you have any feedback, I'm all ears.

UNIDENTIFIED SPEAKER: Thank you very much. Any questions? There is one in the back over there.

EVAN HUNT: Evan Hunt from IFC. Not really a question, but an answer to a question you asked at the end. I really hope that you continue this work, and in particular the feature that I think that you need that you don't have yet is, variable scale cache rate, because it's, and the testing that I have done on our resolvers and others, that is a really significant factor. But it's very nice work and I'm glad that it's out there.

ALEX ROUSSKOV: But it probably doesn't have the cycles to participate in that project, right?

EVAN HUNT: Let's talk offline. I don't think I can personally contribute to it, because I've had access to commercial code that covers some of the same area, and I'm afraid that it might be intellectually polluted. But I'd be very, very happy to see this progress.

ALEX ROUSSKOV: Okay. Thank you.

UNIDENTIFIED SPEAKER: This is the last question.

FRANCISCO: This is Francisco [inaudible], I work for Inner Registry, but I used to work for ISE, and we were building… I know that this is really a hard problem.

I've been doing this for a while, specifically in my group. And one of the things that we notice is that when doing the TCP testing, it's actually harder than people think because most of the time you end up testing the client and not the server.

So, among other things, we had to develop a little piece of software. I know that friends from JPRS, they initially developed, perhaps for testing or as a lab experiment, they developed a TCP testing tool. We couldn't use it because of how, the things that we needed to measure were different from what they were measuring, but on the effort, we ended up building a small TCP DNS testing tool, that I believe that it has now been bundled into bundle 9.10, if I remember correctly, I think it's 9.10.

The TCP DNS performance tool? Yeah, so it's available [inaudible] directory, and it's not perfect, but it does a pretty good job testing TCP performance on the servers. Of course, basically uses, has the ability to reuse sockets and a bunch of other stuff. But definitely the approach that I've seen, which is what we ended up doing was, what you consider the swarming mechanism where you have multiple computers testing server specified time, and then you have to aggregate all the results with the script, to actually get the real results.

So, we can talk offline. We have some ideas on what we can do. We will definitely love to use this tool and test some of our own servers. So.

ALEX ROUSSKOV: Sounds good. And I agree that TCP testing is a big problem there, yeah.

UNIDENTIFIED SPEAKER: Okay, so there is hope at the end of the tunnel.

[APPLAUSE]

UNIDENTIFIED SPEAKER: All right, the next project is Jacques, running bake-off of DNS servers, and I leave it to him and ICANN [inaudible]. This will deal with Christina.

JACQUES LATOUR: Okay, I'll start while she is loading the slide. I got myself in trouble of running a panel on DNS server implementers, because I wanted to know what was going on in the landscape. I think three years ago we had a session, we had a couple of people, I see [inaudible] and somebody else did a presentation on the state of their DNS server implementation.

And throughout the last couple of years, I have seen ad-hoc presentation here and there on various topics of name server implementation. But what I wanted to do is I wanted to know where we were at in the landscape. There is a lot of different participant. There is not DNS. There is NSD, default, binding [inaudible], power DNS, Microsoft. There is a lot of people.

And what I wanted was each one of them to spend 15 minutes, uncover where they're at in their product maturity, where they were living for small and large ccTLD operator. What they're working on, like the latest development and why. Below the bit, or they're developing their server to be more performing or more functional, and what is important…

And what is more important that… Is it more focus on building performance application? Or is it more about having a functional application that supports all of the RFCs, and more features in the back for administration reporting and all of that? I wanted them to talk about a little bit about the decision process for the future, because we've been asking for new features, and we've had different response from different implementers.

Some are focused on different aspect of the tool, and that might be guide to we're a better tool. And I've seen, in the last year, I've seen a lot of slides around performing, but performance, some venders could do, I think, 1.4 million queries per second, some do 100 or 50. So what I wanted to have at the end, if we have enough time, who… How important is performance management now versus compliance in having a good DNS server tool?

If we can spend 15 minutes each, if we have enough time, we can do a little panel and questions.

Can you load the…?

VICKY RISK: Well, I guess I can get started while we're looking for my slides. I did them way ahead of time. My name is Vicky Risk, I'm the product manager for BIND at ISC, and in fact, I'm the entire marketing department. It's a little daunting to talk to this group about BIND because I know probably at least half of you know more about BIND than I do.

But, okay.

Jacques asked me to speak specifically to the requirements of the small ccTLD and gTLD operators, and I just want to say that, one of my primary motivations in coming to ICANN actually is to learn more about the requirements that ccTLDs and gTLDs have. So I encourage any of you to please come and talk to me. I'll be here all week, and let me know what are your requirements, limitations, or new features that you're looking for in BIND, if you're a BIND user.

Luckily I'm not here by myself. Quite a few of my colleagues are here from ISC. It's actually a long car ride here from our headquarters so that helps. I just want to mention, so on the top level Peter [inaudible] and Jim Martin are both working in network operations at ISC. And relevant to the ccTLDs and gTLDs, in addition to running F root, and we're open to talking to people who are looking for F root nodes in their local countries.

In addition to running F Root, we also run a secondary name service that allows ccTLDs to get a broader footprint, perhaps outside of their country. That's something to talk to Peter and Jim about if you are interested. Maybe I need to reduce the size a little bit so you can get the whole slide on here.

There you go. On the bottom line there, Evan Hunt, who just asked a question, he's over there in the back row, Eddie [inaudible] and Michael [inaudible], are all part of the BIND support team. So if you want to talk to somebody about your BIND issues, we have a good representation here.

So, Jacques sent out a long list of questions, and I'm going to try to stick to it as faithfully as I can. I may go kind of quickly through it. First thing

he asked about is, what are primary development initiatives in 2014? As most of you know, in 2014, we decided to discontinue work on BIND 10. One of the primary things we've been doing this year is refocusing our efforts on BIND 9.

Early in the year, we released BIND 9 and 10. And since then, we have just recently come out with the first maintenance release 9 10 1. We have spent quite a bit of time, probably six months, working on supporting customers that were impacted by some of the DDOS attacks that we've been talking about in DNS ORC. And in building some tools, and using some tools that were leant to us by some friends, to simulate these DDOS attacks.

In addition, after the heart bleed was discovered, we asked [inaudible] if they would do some [fuzz] testing for us, and we have benefited from that. One of our other initiatives that is not checked off here because it's ongoing, is improving our automated test coverage. During the years when we thought that BIND 10 would replace BIND 9, we may not have kept up with adding automated test coverage for all of the new features that we were putting into bind, so we have some catch up to do there.

And also, I'm excited to say that we have hired some folks who are going to write some new DNSSEC documentation. We have heard loud and clear that it requires a PhD to get the DNSSEC working. Another initiative or two initiatives that I'm excited about, we have made an effort to reach out to the open source contributors to BIND, prioritize, reviewing, and accepting their packages.

And to communicate better with the operating system packagers. Most, probably many of the BIND users actually get BIND through an operating system package, so we have to stay in touch with them, and do what we can to consult with them and make their life easier. One thing we're excited about, we opened up a public duplicate repository of R get, at source dot IOC dot org. Also, on our website, we've posted a developer guide.

I'm not loud enough? [Laughs] Okay. We re-hosted the DLD site on some newer hardware, and we're also working on operating our bug database.

Of course, I wouldn't want to admit to any missing features, but there are some features that I think are certainly specifically of interest to ccTLDs and gTLDs, particularly in the DNSSEC area. We have key generation, we have inline signing, but we'd like to have a tool that would automate rolling the keys, managing the overlap, and taking care of the housekeeping, deleting the old files.

Yeah, there is a draft related to that, the key timing draft that Steven Morris, our Director of Development, contributed to. There has been a lot of discussion in the IETF about DNS parent updating. We would like to implement the CDNS and CDNS key records. And we have an implementation of a negative trust anchor. It is available today to subscribers but it's not yet in the open source. That's coming out in 9.11.

For most of you, by 9.10 is probably still on the roadmap, even though it came out in April of 2014. The two features of most interest, I think, to ccTLD, or gTLD, or the map zone file format, which substantially sped up

the process of restarting BIND, and support for the native PKSC 11, which simplified HSM integration.

9.11 is planned for the middle of 2015, there are a lot of features planned for 9.11. Several of the DNSSEC features that I just mentioned. In addition, I'm pretty excited about some extensions that we're hoping to make to [inaudible], to enable you to add a zone with a configuration on your master without having to also separately transfer zone files to slaves. We are also looking forward to integrating with the DNS tap project.

I've actually gotten some good use cases from some of the folks here in the room for some of the things that you would like to be able to do with DNS tap and BIND. I'm just going to focus, as I go through these slides, get that guy to go away.

There we go, on things that are specifically of interest to ccTLDs. So, next we have to talk about the software support strategies, the development and support team, and the security vulnerability process. So, the question was, what is our support strategy? And in fact, providing product support really is our whole strategy.

You see in this little pie chart, that giant blue slice, nearly half of all of ISC's support comes from BIND support subscriptions. I know over the years ISC has tried several different models for funding ourselves, but we think that providing a commercial service that is closely aligned to our core mission, is probably the most sustainable way.

I'd like to point out that the maintenance and incremental feature development of BIND is entirely supported by support subscriptions.

We have multiple support levels, with different SLAs, and I'm not even going to let you read that slide. The most basic subscription is 10 grand a year. This is good for people who don't need support, probably many of you feel like you don't need support, but it guarantees you at least three days advanced notice of a security vulnerability, assuming this is not one in the wild, and a software fix for your problem, and enough time to patch your systems, maybe not even work over the weekend prior to the public announcement.

And we think that this is the baseline that everybody who is running BIND, certainly TLDs should have. There is my sales pitch. If you also want support, why would you want support? Well, without a support contract, what most people use is a search engine. And if you have a support contract, you can talk directly to the people who write the software, and hopefully we can be more helpful.

You get a lot more privacy. You don't have to post your problems on a public mailing list, and have everybody tell you that you're an idiot. You can share your config files, or your logs with us, and those will be kept confidential. Also we do put a priority on addressing the bugs that have an operational impact on our support customers. We'll do a configuration review with you on an annual basis. A lot of problems are due to configuration problems.

And at the highest level of SLA, you can get someone to call you back and start working on your issue, with 30 minutes lead time, any time 7 by 24, for a critical issue. We have multiple different trains out at any given time. We publish the end of life dates on our website. In fact the 9.8 train has just gone, been announced to go end of life in September.

So we will only be adding critical bug fixes to that. We have ESV versions, 9.9 is ESV. That means we'll support it for, a minimum of three years.

It first came out in 2012, and at the moment we're planning to support it through 2017. I think we probably support more operating systems than any other DNS server. There are several others that I thought of putting on the slide, and I thought I don't want to make any promises. We do all of this with a very small staff. We actually only have three dedicated software engineers, maintaining BIND, plus a part-time cryptography expert.

And we have a technical support staff of four. This is a very small staff. And we provide 7 by 24 on-call coverage, where someone will get paged for a critical issue. Part of the reason we're able to do this is just by luck, we have people in different time zones all around the world. These are only the ISE staff members that are associated with supporting BIND.

We do also sometimes call on our network operations folks as well. We're pretty proud of our security vulnerability process. It is published on the web. You can read about it up there. The main thing we would like to emphasize is if you find an issue that you think might be a security vulnerability in BIND, we would much prefer to discuss it with you, secure it over a secured communications confidentially, rather than posting it on a public mailing list.

We have, in about 10 places on the website, instructions on how to contact us. And I think Jacques is telling me to speed up. The reason for this is so that we can follow a phase disclosure process, and let people

know in advance of the public announcement of a vulnerability. We give all the root operators free advance notice. We tell the operating system packagers in advance so that they can patch their systems.

And we tell, of course, our subscribers and OEMs. We also make it very easy for everyone else to get more information BIND vulnerabilities via our website, and we encourage anybody using BIND to sign up at least for the BIND announce mailing list, where you will get announcement of vulnerabilities as they're made public.

Jacques wanted to talk about performance versus functionality. Most of you are familiar with the history of BIND. It is intended as a complete reference implementation. And we do have a comprehensive feature set, and we are focused on a faithful adherence to standards, and sometimes that does take precedence over performance leadership.

So, with any software, the more features you add, over time, you slow it down. And everybody wants their own specific feature. And so what you have to do is periodically, you have to optimize. Maybe you add new methods, or streamlined code that was already there. A couple of example of this that we did in BIND 9.10, include that map zone file feature I talked about. The DNS prefetch. Some of the work we did in statistics. This is the case with any software.

One of the things in particular that we look for, when we think about performance is really unexpected changes in performance from one release to the next. Jacques said we should tell you what we think the TLD requirements are. Believe me, if you have more requirements that you want us to know about, I really came here to learn about them.

Manageability and stability, I think are equally as important as performance, in particular an efficient automatable process for adding zones and updating a large network of slaves frequently, particularly for the larger gTLDs, or new TLDs. DNSSEC operational support, some of the features I mentioned on the roadmap, in addition to in line signing, and of course, support for hardware security modules.

These are all important to TLDs, for performance the thing that we here about is, the time to incrementally sign large zones. If you want to have a large zone, you need to sign and transfer, you want to be able to do it in chunks. And of course, everybody wants fast reload and restart time.

Of course, we would love to see some independent benchmarking, in fact some of the other DNS server vendors have done independent, or have done performance benchmarking in the past. I'm always interested in that. One of the things that would make that most useful though, is if we could have active participation from users in coming up with some realistic test scenarios.

This would make it much more valuable. I think it would be very useful to users, also if you could compare different configuration options, and do some, get an idea of, if you changed your model for provisioning or something, how that would affect the time to update. Comparisons between different successive versions of the same product, is very important. I would love to see that.

And I think for users, comparison between hardware operating system platform choices per product, what kind of advantages should I see from migrating would be interesting. How do we make new feature decisions? Our first consideration, of course, we have a large install

base, is do no harm. We do try not to break anything. We have a longstanding commitment to open standards.

One of the reasons we come to these DNS [inaudible] meetings, is to hear about the research that Jeff Houston and others do about issues with the scalability, and the efficiency, and the security of the Internet. And we try to contribute to improving those things. Whether or not the request or the person asking for the feature is contributing somehow, that means, are they providing software to the rest of the Internet?

Or are they contributing on bind users, helping other users? Are they financially helping to support ISC? We have to balance the needs of a lot of different kinds of users as well: Enterprise, ISDs, TLDs. Jacques said, "Well, say something about the install base." Well we have no idea how many people are running BIND. This is a graph from the Measurement Factory, it's from 2010.

It's, I believe, Dwayne [inaudible] FPDNS tool. He has told me that he doesn't plan on running this test anymore, because it's not very accurate, and it's getting less accurate to use fingerprinting over time. But in 2010, there were, apparently, at least a quarter of million BIND servers out there, they're probably aren't any less now. So we're probably still at least in the top 10.

I know it's considered bad form to ask a root server vendor what software their using, because all of their operational details are private, but it's a good assumption that F root that ISC operates uses BIND. [Laughter]

So I am on my next to the last slide. Jacques asked, "Are there TLDs out there that are using BIND?" I asked some of the TLDs that are actually supporting us if we had permission to use their names, and these folks did say that we could mention them. I'd just like to point out, any of you here are using BIND who aren't supporting us, if you see people from one of these companies, you can thank them because they're supporting the maintenance for everyone.

The folks from [nas] asked me to mention, and I think it's true of probably quite a few of these vendors, that there are multi-vendor. This does not hurt our feelings. If you want to tell us that you are running a heterogeneous system, and you're running multiple different DNS software code bases, actually that's a best practice.

And in fact, just in closing, I wanted to mention that part of the issue with open source, is that actually a very small proportion of all of the users support open source. The ccTLDs supporting BIND make up about 10% of our support base. One of the initiatives that we've been talking about, and will be talking about some more this week that we'd love some input on, we're talking to [inaudible] Labs about how we can collaborate, so that we'll both be stronger organizations and so that we can provide better support to the community for our DNS servers, particularly heterogeneous installations.

That's the end of my comments, and I put some references for people who are looking online.

JACQUES LATOUR: Thank you. That was good. So, you said you had some sort of a joint venture with [inaudible]…

VICKY RISK: We're just talking about what we could do, what are the opportunities. We don't have anything to announce yet.

JACQUES LATOUR: All right. On that topic, we'll go with Jaap from NLnet Labs.

JAAP AKKERHUIS: And I'm Jaap Akkerhuis from NLnet Labs. And we're even smaller… We're working on getting my slides up. And we're even smaller than the ISC, we don't even have a marketing department.

I was not quick enough to step backwards when the slides needed to make. So that's why I'm not doing do this. Also [inaudible] has a big list, but [inaudible] time, so we decided to ignore his list, and just tell about what we're doing with DNS. Can I get the next?

I tell a little bit about our mission, because we're not just DNS peddler, but we do a lot of stuff as well. What about history and things where we come from? And the things we do with DNS, and especially keyed towards TLDs. What our support is current plans, and little bit about open source in the end, to finish it off. Next please.

And this looks interesting. [Inaudible] to these slides. But basically, we have way more broader mission than just doing DNS. I mean, it's basically to provide globally recognized interfaces and expertise for

those technologies that turns a network of networks into an open Internet for all.

Well, this is the reason that we are followed by [inaudible], which is actually for non-profit organization, with special status of ANBI status, for tax purposes. And the whole… We actually live by donations. We also, the endowment foundations promised us for support for a couple of years, but slightly we are running out of this promise. There is another two years to go, but we really [inaudible] for different source of income for long term stable base. Next please.

And where we come from, we come from a very strong Unix background. And that means that it also kind of shows in the way that we actually attack problems. That kind of works.

And so we have really the tools based solutions. Well, basically the idea is that you do one job at a time, and do this signal job well. And also, you don't really want to make too much policy decisions for the operators, because everybody is running a different shop. And what we also want to do is to show that, do reference implementations, and show that they actually work. So we're really following what's happening in the IFCs and that's a basic where we get all, what we do, which features we implement.

We try to do everything which is a full, more or less, an excepted [inaudible] status. We don't want to do an experiment. We also know that people have special requests, well sometimes we do these things as well, especially if you find an useful, and also we are able to actually spread around the [inaudible]… very [inaudible] ways of saying, I wish

you'd work with them, and often it's too much work for just too little [inaudible].

But they also want to do is not just show that you can do a reference implementation, but really industry quality [inaudible] industries, probably be between calls, but actually quality is high in what we are doing, and so in the end, there are also things we don't want to do because we don't think it's useful. Also, there is not enough people asking for it, and we have to prioritize with a small amount of people of what we do.

And the next please. The [inaudible] relevant for this company, and well, the first thing is NSD. It is an authoritative name server, which since we're talking about TLDs, they probably main purpose is to authoritative name servers and not recursive. And the other thing is LDNS, it's actually a library to manipulate a DNS data. And there is tools based on… And a couple of tools based on LDNS as an example. And sound pretty useful, which is kind of a [inaudible] replacement.

The key generators for DNS and things like that. This is [inaudible] quite some tools, so I'm not really… They're more examples for what you can do with LDNS and others. The other thing we actually took up because it's kind of hanging, being orphaned somewhere, is to take responsibility for opening DNSSEC. And open DNSSEC is actually all of those things that you at least do to maintain his DNSSEC keys, and to [inaudible] …if you think they're useful.

And so this is… That's actually different part of DNS, which you can actually do separate tools, like we said before. But let me go a little bit about the history of NSD. You can see what's really, how these things

are happening. And the first version of NSD is that we had all the answers, precompiled. I mean, everything you always wanted to ask, we already had the answer first.

So [inaudible] in and out again, just serve it, no complications, nothing like that. I mean, that's all, and there is a special build for the root servers. And the root server operators found, I mean, all the software that they were using, they were slowly migrating to the same version of software and [inaudible] another one as well, just in case… So that one showed different [inaudible].

For security. So and the root zone has [inaudible] precompile [inaudible] for the root zone, because at that time, it was 2 60 TLDs, that is it, it never changes but once a week maybe, and so it's… This is very small. And things started to change when DNSSEC [inaudible] to the pool, and because it has specific requirements for [inaudible]… Things need to be sorted and you need to do all the stuff, and signed negative answers.

And also, because the success for the root, the other TLDs were asking whether we couldn't help them out because, and because I mean, having everything precompiled to memory, each memory. And so can't we do something about the memory [inaudible] and even be as fast as this. So that was somewhere halfway [inaudible], and then the ISPs came to us and said, "Well, we do [inaudible] days for service for the customers, and that is different ballgame."

Because there are lots of little songs you have to serve, and then one big song. So version three, added song support, although not really hard, and starting up is always a problem. And it was still eating a lot of

memory. So now we have the latest NSG 4, and we actually completely back to from where we started in doing the oversight. I mean, now things are actively, the way of how to do all of the [inaudible], and all the stuff you need to do.

It turns out that you can actually build a fast server without eating too much memory. It still eats quite a lot of memory, but memory footprint should went down, and although you could still do electrical pile, everything and keep it in memory. You can still do that, but the [inaudible] you have the speed is not that big anymore. So, that's…

It helps in getting, doing a lot of songs, have a quick change some day, things like that. And we still serve the root server, a couple of root servers, very nicely. So this is kind of the way we develop things, depending on the [inaudible] and we change architecture. Next slide please.

LDNS. LDNS started out as library to manipulate data. And some tools using the library. And the whole idea of, inspired by Perl NET:DNS. I mean a lot of TLDs are using that in the internal system for doing fairly [massaging?] of the data and doing that with big, lots of data. Perl starts to show that it's really interpreted and so we did kind of a re-implementation of all of the pools in [inaudible] or it's now getting really speedy and it's actually being used for a lot of things.

Interesting thing is that people have this [inaudible] and Perl [inaudible] to receive [inaudible]. So it looks like we split back, but it's different then doing it really Perl because it's bind to the C library, and it's different interpreting of Perl. So it's still gain some of that. It's kind of funny that we come back to where we started. Next please.

If I said we are not really principally, I mean, software building, DNS servers, and we do other products as well. [Inaudible] because of cashing of validate name server, and especially used by ISPs and by other things like that. This, we're working on Engrid, and it's complete different thing that has to do with PCP and PCP policies. And Engrid is actually, not really to for doing routing, but routing policy engine.

So seems that you want to do, the policies seem to be routed for that. We work with other parts [inaudible] Verisign, getting a new API into DNS itself. And so application can actually have more insight, it's what's happening in DNS, notably for DNS [inaudible] things like that.

And Perl Net:DNS tools for getting orphaned as well, so we kind of maintain them as well. And that's a couple of the things we do. Next please. And we still, just a couple of random ideas we start [inaudible] around, which are some things kind of work, but actually useful. I mean, a lot of the things we play with is master don't is what we call it. And which is more provision engine for NSD.

I mean, as you have noticed, we are read more on [inaudible] what we have is more the [inaudible] thing, and not turnkey solution, but that's more the style, bind with this [inaudible] for you, including signing, and key management. So the other side for the people who want it.

And the other thing we have to play with is actually out there and in use by some people is Key DNS. And this, so before you ship your zone file to the server, and you actually do some live tests to see whether or not you're proficient [inaudible]… You saw a file on the way, and so that's, we work on that off and on. Next please.

The other thing is really what do we do in support. Well, set small group of specific hand holding and find people [inaudible] file end zone. This is not really what we're good at. But certain parties are doing it. [Inaudible] is offering support for [inaudible]. And there are people doing actually DNS courses, and using NSD, Inbound and BIND, the things they teach.

And for a more direct higher level support, and we have secondary scope open [inaudible]. And basically the idea is that they actually do the contact work without [inaudible] to find this out with the tax man, and they can make a perfect… Actually [inaudible]… doing this app for it, and so if you have specific wishes for support, talk to him.

Same with specialized services. If you really want to have some done and built in as an extra, then [inaudible] is the one to talk to. Next please. For NSD future plan at the moment, we're doing… I mean, we're not doing a lot of [proficient?] stuff, so we only take XFR in. But people are asking us all of the time, how to do outgoing XFR as well.

Well, since we're not compiling anymore, I mean it's actually an easy way to do it then do have [inaudible] we had before in memory. So we have probably come to do that. Also better interface with open DNS, of DNSSEC. So if you want to do more or less online signing, then together ultimate signing, so it will be built so you can build your own turnkey.

Another thing is trying to see what we can do to facilitate lots of PCP streams into the server. We have some ideas about how to do that. So we come, but just as I see, we have this open source dilemma. Everybody wants to get free support, and then free as in free beer. But, you know, beer barrel is sometimes empty.

I mean, barrels dry out and [inaudible] and things like that. So we are looking at long time support to keep this going on, because what's really happening is we're doing it one time, it's easy. One knock off, but having a long term support, if just following the RFC with new features. I mean, it takes effort and it has its costs. We are always looking for an interesting way of long term donations. Next please.

And if there are any questions I am happy… After the questions is some online references to the details, that I didn't want to bore you with all the details.

JACQUES LATOUR: Perfect, thank you. Next, Peter.

PETER JANSSEN: Thanks Jacques, while these slides are coming up, there they are already. So we don't have to make up some stuff to talk about. My name is [inaudible] according to Adobe Connect. [Laughter] Is that something we can fix?

No, oh yes, there I am. Cool, thank you. So, I'm Peter Janssen, I'm a technical manager with [inaudible], being the .eu registry. So in a sense, we of course, are bigger than Yaap's company, that's called like that. But if you're looking specifically at the stuff, we're actually a relative small group that's working on that, but I'll come back to that in the next slides.

So we have a ccTLD registry. So why did we get into the business of writing a name server? That's basically the first question that might

come up in your head. And basically, when we looked at this, we basically came to the conclusion that we wanted an alternative choice to, the big players that allowed BIND [inaudible] that we started looking into this very obvious candidates to run as a ccTLD operate the DNS.

And we thought that, like, you know, okay there is BIND, there is NSD, but implementation diversity is actually a good thing, and we actually wanted yet something else. So we decided well, let's build something from scratch. Do not look at any of the other implementations, only take the RFC's as the input, if you would call it like that, and start building from scratch.

The requirements that we had at that moment in time is basically, it had to be a public authoritative slave. So we had to be able to take any of the binds or the DNSD's that we had running with [inaudible] and replace it with a [inaudible] for instance. So, talking about standards [inaudible] in and out. Notify being sent obviously, and [inaudible] to make sure that the transfers of zone files are actually securely transferred.

Interoperability being the main [inaudible], it was [inaudible] to replace everything with [inaudible] file, but to do a mix and match which is, again, common practice. You will mix and match with all of the others, and make sure that you can have the binds mastered and the [inaudible] slave. That's basically what we're talking about here.

Loading from zone files, resource [inaudible] statements in a zone file, all sorts of standard things. The biggest requirement we had is that we had to support a large zone, and it had to support a high query load, so really, up in the TLD levels. Obviously, DNSSEC had to be there, yet

again, the stand of things like N SEC, N SEC 3, and some algorithms that are mentioned there.

And what the initial requirements, currently [inaudible] has evolved a bit. We have implemented ACL so you can do ACL, saying I don't want to get something from this IP address, or from this set of IP addresses, in terms of transfers of [inaudible]. In the meantime, [inaudible] can be the master so it supports dynamic updates, it supports DNSSEC online signing.

It hasn't had have seen a lot of testing on the DNSSEC online signing parts, what is most notably missing is the automatic key management. This is still something that we're contemplating how we would invest solved that, but it's there in its infancy, let's call it like that. There are a set of tools to do DNS queries, to do dynamic updates, to do remote control of local [inaudible] remote control running [inaudible] for instance.

And what is interesting for us, at least, is basically there are all sets of libraries that you can use to build up and to do other projects. One thing that we're doing internally, for instance, on the one hand we are a registry, so we have a database that contains registration information about domain name and some technical aspects like name service and keys, DNSSEC keys, another on the other hand, you have a running name server with a zone, how do you do this [inaudible] mismatch between a sequel database and a zone.

What we have built is something called a dynamic updater that basically listens on a message queue, gets information from the registration system, extracts the interesting information, and crafts dynamic update

messages after [inaudible] DNSSEC pre-checking for instance, it goes and checks all of the name servers, to see if the zones are currently signed, keys are correct, and so on and so on. And then it crafts DNSSEC, so dynamic update messages and sends it on to the hidden master that will propagate to all the public slaves.

That dynamic update thing is basically something that is built on top of [inaudible]. The whole idea is that it is an useful set of standard libraries that you can use to do some sort of application that it needs some sort of DNS [inaudible] let's call it like that. We started this around 2009. It's not very clear, I tried to find my emails, the exact date, but I was unable to, to find an exact date, but it was somewhere around 2009 that we started thinking about this.

I had this geek in my team that, you know, wanted to do, suicide himself it wasn't in C, so we decided on C. We came pretty quickly to the conclusion that open source and basically a license that said you could do anything with it, as long as you give some credit back to us. That's basically what a license was all about. One important aspect as well was portability, we had an earlier presentation where there was a claim that they were running on the most different operating systems that were out there.

I'm not disputing that, but we're running on a lot of the operating systems as well. Most [inaudible] missing for the moment is Windows environment basically, because we haven't had any need to test it, but basically it should run, it hasn't been done yet. As I said, we had on the one hand, a bigger company, and on the other hand, a smaller company. If you look specifically at the [inaudible] efforts. There are

two developers that are actually working on the [inaudible] core, and all of its things around that.

But a part from that, of course, we have a testing team that is responsible for amongst others, testing the [inaudible] registration system for the dot [inaudible] registration system. We have another 18 members support team, we have the web team, all those people chip in to do some work on the [inaudible] format from time to time.

As for support, we have no formal support contracts in place, and I've put in between brackets yet. If there is a need, if somebody actually wants to talk to us about, you know, I want to pay you some money to get some guarantees about [inaudible] features or whatever, please come and talk to me and we're most happy to work something out.

For the moment that is, the mailing list, that is the website, that is direct contacts with people that actually are working on that, anything you would need to get you moving forward. What is very important is, first and foremost, [inaudible] is an authoritative public slave, but we are very willing to add useful functionality, as long as the performance stays okay for some definition of okay.

In the beginning, we have focused a lot on performance testing. In the meantime, we still do that but we just checked that we're high there, and we're talking about a few hundred thousand queries a second, on the run of the mill hardware, and that's just good enough. So when we had functionality, yes it could be that the performance goes slightly down, but then it's slightly going down, the whole idea is it shoots the, foremost a public slave that is, you know, fairly performing in terms of enhancing [inaudible].

Another important aspect, as you add [goat], that is [goat], it can be [inaudible], it can be abused, it can be whatever. So a level of security has to be maintained to make sure that whatever you add can't be abused, in its role as a public slave. And we have done recently, for instance, is the last thing is have compile time options, where you actually can say, you know, I want this part of the [goat] not in the binary.

One of the things that we have added for one of the groups of people that we're talking to is a functionality where a compile time you say, "I want it to be a public slave." And it will just not compile in the DNSSEC signing, it will not compile in dynamic update messages. It will not compile in a whole lot of things, it will just take the bare minimum that is necessary to actually make it as useful as the task that you have in hand, in that case a public slave.

Obvious to say, if [goat] is not in the binary, it can't be abused. That's the whole idea. And if you add new functionality, it's always, it is always the idea to be able to say, "I want it in at run time, but also at compound time." What is that useful functionality that we're looking at? In the meantime, we have added the possibility, you change the config file if you define, you can tell it's either remotely or locally via this remote control tool, to reload and reparse the config file and actually online do whatever is necessary to configure it again without stopping, answering inquires on the things that it is doing.

One of the whole topics that we have been talking about, why the bid is what we call dynamic provisioning. And the whole idea is that you use the DNS protocol to actually remotely configure a set of running name

servers. So the whole idea would be the only thing that is happening between a master name server and a slave name server, is the DNS protocol, and not just using the DNS protocol to do zone transfers, in that case, but also do, what we then call as a maintenance zone transfer where the maintenance zone actually contains the configuration of which zones that a slave should be slave for, where this master is, what the ACLs are, what the [inaudible] keys are, whatever is necessary to actually configure a name server to be whatever it is that you want it to be in terms of slave or master for your zone.

We have running [goat] in our labs, we are playing around with it. We just haven't gotten around to, you know, finalizing it. But eventually, the whole idea there is you have DNS operators that need to maintain tens, hundreds, thousands, tens of thousands of zones, and every day, hundreds of thousands come in and go away again. This is what the problem we're trying to solve there by using the DNS protocol.

JACQUES LATOUR: Peter, is there RFC's around that?

PETER JANSSEN: No, there are no RFC's around that. We are looking into that, but again we haven't come very far. We're actually writing something down, they might resemble a RFC but, the intention still is to do that. Another thing is, for a moment [inaudible] uses a zone file as a backend, so you start it up and it will read a zone file fitting pretty much like, you know, with binds and others.

It will read that and it will start processing that. The whole idea would be that [inaudible] would directly connect to a sequel database, very much like [inaudible] DNS for instance, or use a message queuing backend where it actually gets information to prevent it from having to listen to dynamic updates or something like that. But this is still very vague and potentially on our to do list.

Same thing with the next topic which is a validating resolver. When we started this, we thought, yeah, let's go all the way. We're not so sure anymore that this, a way that we want to go, and like Yaap says, you want to do something very well, and by trying to do everything, you have to compromise and we're not so sure that making a [inaudible] resolver will actually be a compromise that we're willing to take. But again, it's still very much open.

Set differently, the focus groups have been and always will be, root and TLD operators, and hosting companies basically, lots of queries, potentially lots of zones, potentially big zones, but the added focus on removing and adding zones on the fly without too much hassle.

Lastly, .eu has a [inaudible] authoritative slave for well over two years now. So I think we're ready for primetime, and that was one of the questions that I do want to answer in explicitly asked by Jacques. We have been running for two years, [inaudible] slave, it's answering thousands of queries each minute that we talk. It is very stable. We are pretty confident that [inaudible] as an authoritative public slave for a small or big zone is actually a viable option, and it's out there to be used.

We talked to a lot of people that have shown some interest. We're working with them if they have any requests, requirements. We have more than willing to work with them. We're actually working with people to add functionality, or change functionality, whatever is necessary. And you have a few of the contact possibilities there on the slide. Thank you.

JACQUES LATOUR: Thank you Peter. Is that your new t-shirt?

PETER JANSSEN: No, we have been debating internally if we would restart the series of t-shirts that we have done in the past. We have some nice slogans that we stole from some other people, and they might become… One name server to answer them all is one of the things for instance, that might come up. And now [inaudible] so I would have to come up with something else, but we have a few more, and we didn't get around to do it for this ICANN, but we might do it for the next one.

JACQUES LATOUR: Cool. Thank you. Next is Ondrej, with cz. Talk about knot DNS.

ONDREJ SURY: Hello. My name is Ondrej Sury, and I'm from cz. And okay. So what's knot DNS? This is a set of requirements that we had from the start. Well, it's high performance, scalable authoritative DNS, it's free open source and also written from scratch. It's under very active development, and we try standards, and well very fast implement them.

One of the requirements was also non-stop operation, so we can do the runtime reconfiguration of the server and you don't have to restart it. Well, our targeted groups are also root, TLD, and DNS hosting companies, and I think that applies to most DNS servers there.

We have also now DNSSEC automatic signing, and something we call dynamic modules. This is the quick history, well, and the future roadmap of the knot DNS. Our first public release was in 2011. And after some very active development. We are not at DNS 1.5 version. We have added DNSSEC automatic signing in 1.4. And the last release, we're actually focused on refactoring lots of code. And we were able to reduce the mobile lines for the code while adding more features.

And those features are dynamic modules, and I will speak about them in a bit. And we were able also to reduce the memory usage while pertaining to speed. Yeah, that's better. Just today we have released the first release candidate of DNS 1.6, which will be our long-term support release, because it will be the first knot DNS included in [inaudible] next stable release, so we have decided to go with this version. And there is only one new feature, apart from the bug fixes, and it's persistent timers.

So what are the dynamic modules? This is actually the whole zero request for users. The modules are just small hooks in the query response processing at various places. And we have, right now, two modules. One of those is sanitized resource records, it's, that's the thing that's developed on a request.

And it's use for dynamically generate the PTR records for IPv6, because well, you just can't have the [inaudible] in the config files, for example,

for DSL lines, so you just create a rule how to resize them and this will be done automatically by the server for [inaudible] records. The other one, a module that is currently done is DNS tap query response logging, it's structured binary log from [inaudible].

It is more possibilities like split horizon, Geo IP. You can do poor man's high availability, unbalancing, and stuff like that. The persistent timers, this was requested by Ripe NCC. And we now store the timers for expire, refresh, and flushing the zones on disc. So they will survive the server restart. Just a little bit, looking into the future. I know DNS 2.0, which should be released summer, around the end of the year.

We will have improved DNSSEC support. We are going to switch from open SSL to GNU TLS. It's, well, we have decided to do that even before heart bleed, because we think that all DNS servers out there are currently running on open SSL, and we want to add more diversity to that. And it will also, hopefully, improve the support for hardware [inaudible] modules, implement PKCS 11.

We will also add key and signing policy and tools for key rotation, and well you have heard about it from all of us. And also, well online or inline signing which will help us to do more fancy stuff like minimal NSEC 3 encloser, adding more security to NSEC 3, and signing the output of dynamic modules.

And we well, we are talking to support and it looks like that the configuration file is evolved to be more machine readable for big deployments. So it's also going to happen. Next year, well a lot of things we would like to do. While doing the dynamic modules, and the hooks in the code, we actually discovered it. It's not that hard to add

different backhands for storing the zones, so we're really add some key value databases like the memory database, or SQL databases, because that's what we asked from time to time.

And for really big deployments like, well if you have millions of zones, it's just not feasible to try and config file, text config file, so we really only have an option to load the zones from database, the list of the zones from the database for such big deployments. And also one of the things that people regularly ask from time to time is that the [inaudible] API, so they can block it, not DNS into their existing systems and do some stuff, rest or how do you call those fancy stuff?

So it's also one of the things that we would like to focus on. There is also one thing we are developing right now, it's resolver stuff. Actually it works now, but it doesn't have all of the stuff we want to have by the end of the year. It will also, dynamic modules that could layer on each other. It will, well the focus it, persistent cache. And we are also trying , well if you're following that stuff at ITF, so there are talks, how to add more privacy to DNS, and one of the techniques are [inaudible], so we will try that, because we can right now, because we don't have any users because it's not released yet.

Licensing, we, well, we pick because it's our mindset, the GPL license. And we try to be as open to this as possible, and we think it's a good thing to do so while we have the mailing list, it's obvious, but we have open gate repository from start, where you can follow the [inaudible], unless there is really, well security about we were just, that needs, well, serious handling, then we will publish all the code as we do it.

While speaking about, there are two ways, best one on mailing list, and we consider our users to be our friends, so we generally try to honor direct requests. And there is also an option to have contractual support over email, custom email or phone, but it's well, it's in its infancy. So, if you're really interested in that, then you can speak to us about that.

For the support and security disclosure, we just had a, well a crash, a bug in the code, so we are starting out procedures, so we have requested the [CD?] number, and well the process right now is if we know you, we will let you know. As I said, we consider our users to be our friends, so we have informed, I just think that users should know that there was a box updates to patch their servers beforehand.

So if we don't know about you, you are like, should try to also know about you. But we hope we will not have any more bugs like that. So, one of Jacques questions was performance or functionality. And we think that both are important, and you don't have to sacrifice one for the other. So, while from a performance perspective, you need to sustain a high load while under attack, because well you might have noticed that there are attacks on the DNS infrastructure right now, in the past two years.

But on the other hand, the functionality is also important because well, we [inaudible] and it's our files are [inaudible] DNS support is a must, especially from the interoperability principle, you need to support all R types, even the ones that are not commonly used. And it also helps with the deployment, well, both new deployments and existing replacements of the other DNS. And we also try to follow the

robustness principle that you should be very current on what you except, and very straight on what you send.

Also, before the benchmarks, I think one of the things that we have started, do the public benchmarking, and we still think that the benchmarking should be as open as possible, so we try to publish all the code we use. We published the hardware specification. Recently, we have done testing for network cards, because well, they were 10 gig network cards, because there could be a big difference either in the hardware, or in the drivers, or in the operating systems.

So it's something you definitely need to look into, if you are planning well [inaudible] for months, it's not just the software, but you also need to test different network cards. So for benchmarking, definitely the hardware specification is very important. Also there is some stuff you can do in this operating system, like the configuration of the network, parts of the configuration, how to interprets are distributed, so that all needs to be documented.

And while basically, the one thing is to test the software SA when you get started from some package, or if you do some tuning. And we think, and we have been calling for that, for a long time, that it really should be a collaborative or at least down by some independent party, and we all, all the windows should participate in that, so we are not comparing apples to oranges.

As for the new features, there are several channels. We're accepting new features, it's the internal users because we are using obviously, because we are dot .cz registry, well our, network operators request some stuff. We also accept requests from well external users, and we

apply common sense in all of those requests, so just because, well it must make sense basically. We also follow what's happening in the DNS community, it's why we are here.

We followed the IETF process, and we actively participate in this. And we also follow the non-IETF idea, like the RRL, the resource rate limiting that came from policy, or I really did like the idea of NSEC 5 in the [inaudible] presentation. So we will be looking to that. So and for the existing TLD users, we are using, at CZ NIC, it's approximately one-third of the .cz servers are running knot DNS.

.dk has one slide running knot DNS for, I think, two years now. And also one third of RIPE NCC DNS servers is running knot DNS, well both RIPE NCCs, I think, are [inaudible], and it's more notes than just one, and it's hosting 77 TLDs and some other interesting domains, so it's also on knot DNS as well.

And that concludes my presentation. So thank you.

JACQUES LATOUR: All right. Thank you, that was good. So we've got 20 something minutes left so we're good for two more.

Next one is Microsoft with Ralf. Nominum.

UNIDENTIFIED SPEAKER: It is Microsoft here, presenting. 1,2,3, no.

UNIDENTIFIED SPEAKER: So it's the Nominum slide from Adrian.

ADRIAN: So while we're waiting for the slides to show up, I'm Adrian, I'm with Nominum, and some of you heard me speak yesterday was Ralf Weber. We both work together at Nominum. And I wanted to thank Jaap from NLnet Labs, first because he was the first guy to put his slides together, didn't follow your process exactly, because ours is the same.

But to answer the questions that Jacques put to us, for licensing and support, it's a commercial model so it's different for anybody who has spoken so far. Performance versus functionality, yes we do both. For what drives new functionality, the short answer is the market, so it's similar to the previous presentation, where there is internal user requests and external user requests.

We balance off both in conjunction with supporting the commercial model that we're part of. For functionality, there is a number of pieces of functionality that we're going to talk about today that I think are important to TLDs, and these are in no particular order. So these are large zone support, high query rate with very stable latency.

We do DNSSEC very well. Multi-mastering, we have that capability out of the box. Data collection, so we can collect data on the fly and we can aggregate it to a central point. And we've also got management APIs, so anything that you can do from a config file perspective, you can do with an API, and you can do it in Python, or Perl, or C. If you want to do it from a command line perspective, because we're used to network device, you can do it that way as well.

And to start off, some of the things that I just spoke about earlier are listed on the slide. And it's sort of a bottom line up top, so it's a list of all of the things that we can bring to the table. So proven high performance. The numbers that we publish are about a quarter of a million queries per second. And from a scalability perspective, we tested up to a billion in resource records.

So it's a large number, and we're deployed in carriers and very large enterprises as well that have some special needs around VOIP. From a security perspective, if you find a vulnerability, I'll mirror what the ISC said, let us know. But we haven't had any CVEs with off serve in 10 years. Always on service is an important one. So the configuration changes or adding data don't require zone updates.

The multi-mastering ties in to the always on service. So we're able to deploy in pairs, you can provision against both. One goes away, comes back again. It self-heals. The API's that I talked about are certainly a function of any of the engines that we have off serve, it's no different. I'm getting a bit of feedback here.

We also have configuration templates. So you can configure zone configuration data for allowed transfer also notifies, and you can set that across the board for any zones that you'll create with these configuration templates. We've also got data templates, so if you want to set SOA information or name server records, it's a single point that you can do the configuration and apply it against any zones that use template.

We've also, at the end of this, network visibility and event awareness, and that's tied to the fact that we can do data collection. So we can do

data collection, and based on configuration that you'll apply to the data collection we can notify either with us in MP or with events.

So this is where we were supposed to bring our guitars and sing a song. And I left mine in the hotel room, you know what it is? Okay, this one is good. I mean, I can talk to this one. It was a picture of our architecture, and evidently there was some animation in there that I missed. And what it does is it breaks what I just spoke about out into a series of blocks. There is a lot more blocks than what should be here, but we can skip to the next.

UNIDENTIFIED SPEAKER: Okay. Now we're talking about DNSSEC on, these are the models that DNSSEC is deployed today. I guess the most deployed models are the left one where you have sort of online signing at the server, and the right one where you have kind of a hidden mask of where actually all of this stuff happens. Now no matter what model you deploy, I think we have one of the most advanced DNSSEC authoritative engines.

When it comes to actually manageability. And to put a history onto that, before I joined Nominum, I worked for a large European ISP and we wanted to deploy DNSSEC, but we weren't sufficient with the tools out there, and a large US kind of provider also had the same problem, and was also not kind of happy with the tools out there. So we were both known customers and we came to them and said, "Well, we want to have a solution here."

And the solution, I think it was 2008 when we approached them. And in 2009, a solution that I now present, basically went into production, at a

couple of large ISPs, so the principle is that you normally don't care about DNSSEC, because you guys are, I guess, are making money by selling domains. You're not making money by doing DNSSEC.

And most of the people have DNS data, care about their DNS data. It is nice that it is DNSSEC, but they don't actually care about the details of that. And that was one of the principles that we wanted to make the actual data model of our DNSSEC implementation, so that you didn't have to change anything. So we have lots of APIs, you can also extract zone file database or put them in again.

And all of that is kind of totally DNSSEC agnostic, so you work on yoru data, and the server makes sure that what gets served out is DNSSEC. And we do that by kind of putting the DNSSEC parameters into a sort of zone configuration that we attach to the zone, and that hands all of the key administration, it hands all of the resigning. You can roll over the key.

Say there KN, KS, K, you can have all of that kind of, the server does it for you. You just need to tell the server what you want to do. So, one of the slides I had earlier was kind of, a single command to make a zone DNSSEC compatible. And the server then does all of that.

And we do that normally with a model that you have the zone is in the server, kind of signed, so that we do online signing for some stuff, but most of the stuff we kind of pre [inaudible] designs, and then put in the zone. Now of course, we have one feature that makes that kind of very handy…

Our zone data, we don't have zone files. We have an in memory zone database. So when the server starts it goes… And then a zone is kind of sucked into memory. And then general files are written to discs, so that we have also a consistent image of that. And with that, every change that you do to the zone is kind of version. So whatever you do, you can roll it back or forward, or wherever you want it to. And you also can see the difference between the zone.

So, the zone that we have in the database, it's not just one zone, it's kind of the accumulative of changes that you had over time. And signing a zone is just one of these changes. And resigning a zone is another one of these changes. And rolling the key is exactly the same. So, pretty much everything that you can have to do with DNSSEC, just happens underneath. And of course, we do a kind of logging, if keys are about to roll out, and the case of where you bring the key online back again and stuff like that.

And the performance is, it's kind of usually [inaudible] as the same with DNS. So.

ADRIAN: So for multi-mastering, I don't think I need to go over the risks associated with running a single master, but we have a multi-mastering capability, and…

And the nice thing about it is that it runs out of the box, right? So it self-heals, its proprietary protocol between the engines. Right now, we only deal with pairs. So we'll have a pair for multi-mastering. One of the features that we're considering is opening that up to four. It will do

mirrored DNS updates. So any zone data that you've got is automatically propagated. Where we see this deployed typically is from a hidden master perspective, and you'll have a group of slaves underneath.

For the use cases, for this audience, it's TLD, right? Or ccTLD. We also, we run the multi-mastering capability within VOIP deployments. And it reduces down time. So you don't have to worry about having to bring up another master in the case that one goes down.

So for flexibility and extensibility. Management APIs, I guess I'll touch on that one first. As I mentioned in the introduction, there is very little, in fact I think there is only a couple of things that you can't do within the API, and one of them is to start the engine, so you'll have to use [inaudible] or a service start to start the engine.

But I'm working on a deployment at the moment where we're replacing a set of name servers with off surf, and I've rolled out a slave infrastructure for, I've rolled out a slave infrastructure that's going to mirror what's deployed currently with our APIs. What I'm able to do is I'm able to take one of those slaves and promote it to master, and it's literally about 10 lines of code.

So I can take the zones, convert them into a master, alter the way that the data is stored in the database, and start it back up again. And that's all done via the API. The data that comes… A little bit about the API, the API is largely a hash map API, so it's key value pair, it's easy to…

It would be easy to write your own wrapper for it if you want, although we do have versions that are available. The zone configuration

templates, I did talk about those, and I'm going to talk about them a little bit more. Where it's really useful is when you've got repetitive information, and you've got to create new zones, you can template most of the information that's there, right?

So for any of the configuration items, you can create templates for them and reuse them, and if you make a change to that template any of the zones that you [inaudible] that use that configuration template are changed automatically. So if you need to change notifies for a slave that you're adding in, you can do it in a template versus having to touch each zone individually.

For versioning [robots and diffs?] the only thing that we didn't talk about so far was [diffs]. We can do versioning, and we can do rollbacks or roll forwards for DNS zone data, but what we can also do is we can do [diffs]. So if you want to know what the differences are between zones, you can do that too. There is an UI available, that we have a partnership through [inaudible], and if you're interested in that, we can talk about that off-line.

Composite zones. So composite zones are kind of like templates, but what they allow you to do, is they allow you to lunge multiple zones together. The use case for this, I don't know if it's specifically applicable to TLDs or ccTLDs, but in the interest of explaining what it is, it was largely used for [e-nom], so in [e-nom], you would have different datasets when you were trying to look up a telephone number to determine how you were going to route it.

So you would have local number port, so if you're switching between carriers, and what composite zones allowed you to do was take

disparate datasets, and instead of having to look up in three zones or ten zones for a piece of information, you could just look it up in one. And if there were conflicts inside, we had algorithms that you could use to configure what would be a preferred answer.

Go ahead. Okay, so real time visibility and alerts. This is something that I use a fair amount for data collection. And what we're able to do is, wire speed, pull information out of the engine. And we're able to pull out either, you know, effectively a query log, but we're also able to pull out the answers as well. And the functionality that we have permits us to aggregate it a central point in the network as well, so that you don't have to worry about pulling data back in.

Within the technology that we have for data collection, you can create reports that this component will generate for you. So if you wanted to take a look at statistics for a particular zone, you could do that. If you were looking for, you know, queries in this example that came in with the [inaudible], you could see those things as well.

So it gives you a lot of visibility into what's going on at the engine.

I'll pass back to Ralf.

RALF: So one of the steps that we recently did was because of all of these attacks are happening on the infrastructure. We, of course, did all that we can do on our service to kind of remedy the attacks. So we not only have resource range limiting as it's kind of normally defined, but we can do a more fine rate limiting. So you can decide on either a zone, or an IP, or [inaudible] type, or any combination of all of this, and can say, "I

only want to filter this," and if it hits a certain limit, then you either want to truncate it or drop or whatever.

Now, of course, with the recent attack, on the authority side, this has been a bit, it is good that I think you could probably do it per zone, because then you can at least kind of make the right decisions for the zone.

And another functionality that we added, that's not probably does not apply here, is automatic generation of reverse forward IPv6 records, I think Andre said this, that it does not do that. And with that, of course, we keep all our DNSSEC primaries, but of course, if you are doing an online signing, which you have to in that case, because you synthesize [inaudible] response, you have to have at least a zone signing key at every server that answer that query.

And another thing that we did very late, that I did earlier, is kind of slave zone signing, because our solution from the beginning was constructed to have no changes in data mode, so most of our customers didn't use it, so that's probably why it's late from our side. That's it.

JACQUES LATOUR: Perfect, thank you. I guess we have five minutes for questions, Q&A. So.

UNIDENTIFIED SPEAKER: I was just thinking that if you don't have any questions, maybe you can give us some answers. For instance, I heard about some features that I would love to see in BIND from some of the other presenters, and if I

wonder since you had a chance to talk to all the DNS server vendors, would anybody like to stand up and talk about a feature they would like to see? That they can't get today from any of the vendors.

STEPHAN: My name is Stephan [inaudible] from Microsoft. It's interesting to see that, I think two or three vendors talked about the reverse delegation IPv6, and to be able to do that on that fly. I know there has been a couple of drafts tonight, yeah, but there has been some type of resistance to move with those.

So there seems to be a customer demand here for doing those kinds of things. And I'm hoping that we as a DNS community can come to a conclusion so that I can have a master that's BIND, doing a zone transfer to Nominum, and to a not DNS, and NSD and so on, and didn't have some type of protocol where this can be provisioned automatically.

UNIDENTIFIED SPEAKER: Can you point me to…? I mean, I know there is drafts in the IGF always about how you should do the reverse tree, and they always kind of are brought up and then twiddled down because people can't have a common ground on that. But I don't think there is anything on the specific problem, how you provisioned…

STEPHAN: No, there is not. There is a draft [inaudible], a guy from Time Warner Cable, Lee [inaudible], draft [Howard?] something. He refreshed it quite recently, so I don't know if it is expired yet. But he has actually, he

comes up with four different solutions to this, you could do a dynamic DNS, or you scale, or you can do it on the fly, or to some other, you skip it or do wildcards and so on.

You can probably use D names as well. I've seen people doing that. But you know, I think as the community, we need to come to consensus here, so that different software can operate with each other.

UNIDENTIFIED SPEAKER: I tend to agree on that, but I don't think that it will be covered by that draft. It probably needs some new effort.

STEPHAN: Absolutely, absolutely. But his draft doesn't make any progress, and it's just an informative draft saying here are the different alternatives. So what do you do as operator today when you can't go to ITF and get a recommendation on how to do reverse litigation on IPv6? That's what I would like to see. And it seems a lot of customers wants to do this by having name servers do this on the fly, so maybe that's the way forward here.

UNIDENTIFIED SPEAKER: Well, there has been a lot of proposals being made, but this falls under the subject, complications of DNS servers in general. And there has been quite some efforts in doing that, and they all the time die in beauty. Because nobody really wants to do the work for rewriting [inaudible], I mean, everybody wants a solution, but nobody wants to do the work.

And that's one of the problems, is we… That's why somebody comes up with the draft, and things wow, and this, no support of the community. And the other thing is that a lot of people are, have different ways of the internal processes, and it's very hard to find common enough matters so that everybody can accept it. I mean, it's kind of problematic in this atmosphere as well.

You see the same with [inaudible] which is also standard, [inaudible] variance for it, and we don't talk to each other and kill each other. If you really want to do that, I think the gentlemen on my left said something about creating zones, and it's a typical of way of doing that. The community itself would [inaudible], it's kind of [inaudible] find a lot of stuff, what we think is useful.

But there is no common ground of doing this [inaudible]… was actually did some requirements for that, [inaudible] …input, and it was a draft [inaudible] and after that, nobody could react. So, this need for it, which I don't see how…

I spent a lot of time trying to get something done in that [inaudible], but, yes people want solutions but they don't want to do any work with it.

UNIDENTIFIED SPEAKER: A question, or what? One of the things is that we always present GNS server, server that are designed to serve ccTLDs, or TLDs. But so DNS server are also embedded in some device or CPE. And at that point, it might be very useful to have a way to configure, I mean a standard way to push one configuration so that any server that is implemented can

just read the same file, and anyone can just push the same configuration that is interpreted in the same way.

UNIDENTIFIED SPEAKER: But that's still the [inaudible], and [inaudible], if they can spend $10 on DNS implementation, they find that a lot of money. I mean, that's why you find all of these very broken DNS servers in [DC 5] [inaudible]. And not only, are they broken, these things are made to never be updated. They're only made to be short and shown away.

I mean, maintenance isn't there. I mean, SSEC had actually [inaudible]… And there is some improvement. I mean, the foundation of [inaudible] actually supported the [inaudible] or whatever this is common used piece, and they actually support it to put a DNSSEC in it, and hope in the long term that it actually will be done something.

But this is such a [inaudible] market where, it's very hard to… As long as it works with them, it's okay. That's basically bottom line there. And in this case, $15 to…

UNIDENTIFIED SPEAKER: So I just wanted to make two comments in response. So first of all, the automatically generating the reverse point of reference, I would very much like to see that in BIND, it is on my list. I'm not familiar with the different choices of implementation choice as possible, but I am absolutely willing to ask the other vendors what choices they're making and we'll take that into consideration when we make our decision.

And the other point about configuration, I also, I've been in this business, not the DNS business, not necessarily, but I've been a product manager for decades, and I agree that it's very difficult to get people to agree on a common framework for provisioning, especially basically competing devices, but I'm aware that it's important to the DNS community for operators to be able to have multi-vendor implementations in heterogeneous implementations.

And I know that some of you have built systems where you are switching back and forth from one vendor to the other, rapidly, and reconfiguring, and if there are cases where configuration differences are making that problematic, I would love to hear about something like that. That would be a more focused approach to things where it's important for us to have a similar configuration.

UNIDENTIFIED SPEAKER: Okay. So I want to thank all the speakers today. I hope it was an interesting session. Thank you.

[APPLAUSE]

UNIDENTIFIED SPEAKER: Okay. Thank you very much. There is one more. Jeremy Rowley from DigiCert. And then Andre can close the proceedings.

JEREMY ROWLEY: Hi everybody. I'm Jeremy from DigiCert. And I'm going to talk to you about what's going on in PKI these days. How it impacts you and then some of the free tools that are out there that you can use both from DigiCert and other people, to look at digital certificates and see what's going on.

But it has been an exciting year for digital certificates. We have the deprecation of SHA-1, certs, certificate transparencies being deployed as of January 1st. Certificate life cycles are about to be decreased down from a maximum of five years, down to three years.

The internal name deprecation, I'm sure you're all aware of that going on in connection with the delegation of new gTLDs, but that's going to be accelerated. So that as of next year, you won't see any more of these type of certs out there. Certification authorization, a little bit about just heart bleed and various other bugs. So the first one is SHA-1 transition, where everybody is moving away from SHA-1.

It was shown to be compromised as of 2011, but not in real time, so it's a good time to move. Basically the three browsers have announced deadlines to get certificates off of these. Microsoft started, led the pack, by announcing that in January 1, 2016, all are going to have to stop issuing certificates. And code signed certificates with the SHA-1 hash will no longer be trusted in their platforms.

And then as of January 1, 2017, SHA-1 for SSL will be deprecated as well. So everybody is going to be away in the next couple of years. So Mozilla also announced in early 2015, they're going to show security warning for certificates expiring in 2017, and then in Firefox release, it comes out in 2016.

It's going to start showing untrusted connections, if you have a SHA-1 cert, that expires in 2017. And in 2017 all certificates that have SHA-1 will be considered untrusted. So it pretty much mirrors the Microsoft timelines. Then finally Google came out with an announcement recently that said September of this year, their early release, the carry release, is going to show mixed content warnings, the SHA-1 certificate expiring in 2017.

November 2014, so this year, next month, you're going to start seeing a mixed content warning for certs expiring in June of 2016, or beyond. And then finally in about Q1 of 2015, you're going to see a warning for all certificates expiring in 2016, and even having an interstitial where you're going to have to click through to get to the site for anything expiring in 2017. It's going to show up as non-secure.

So if your customers or you have not switched over to SHA-2, I recommend doing so. There are some compatibility issues. For example, Windows 2003, without the lead cert pack, it doesn't work. China, we've seen a lot of people in China pretty upset by this, because it's Windows XP service pack 2 primarily there, which does not support SHA-2.

But yeah, so everybody is going to have to move. That will finally kill Windows XP probably. So one of the things that we have to show you, well if you can get it from a website, it's a free tool, it also has an API access that you can download. It will go and detect all of the SHA-1 certs that you have out there. You can just scan the domain, and see exactly when you're going to start seeing errors in Chrome. And you can do these in batches, so if you want to find all of the ones in your, if

you're a registrar, if you want to find out ones you've got, you can go and scan those and see what it's going to look like.

Certificate transparency. We've been working on it a long time. It's a Google led project. And basically the goal is to log every certificate issued into a database that is publically searchable so you can see, it gets rid of the whole problem that you don't know what CAs are doing, and where we're issuing certs. In this case, it's going to start with EV certificates and be mandatory in January.

The goal is to provide faster remediation so if you detect a mis-issuance, you can notify the CA and get it taken care of right away, plus it will detect any compromise in the CA like the [inaudible] event a few years ago. The number of logs they have to log this into is dependent on the life cycle of the certificate. Right now there is four logs that are trusted to run by Google, one run by Matt Palmer and one run by DigiCert.

That these are going to be logged into, and all of these certs will be in there. So monitoring tools are being built, so you'll be able to search the log for all of the certs on your domain names, or any domain names that you want to look at and see what's out there.

So and this is what it looks like in the thing, so if you look at Chrome, here on the right. It just says that there is going to be transparency information listed there, it will be an indicator in the UI that says whether or not that certificate is logged properly. And you won't see for the indicator.

Now this is a good step. We want to get to the goal where every certificate is logged, so you can tell what's going on out there with

certificates. The next one is short list search, this is a new project that we're excited and that we need more participation from people who are using certificates or people who are interesting in certificates, to chime in on.

And they're going to be issued with a 48, maybe 72 hour validity period. It gets rid of the revocation problems that have been previously discussed, because once the certificate life cycle expires, it becomes essentially a non-trusted connection. And it's nice because if you were for Alex [inaudible], he talked about how certificates are getting larger, but you strip out a lot of that information that goes into the certificate with the short live certificate, because you don't need an OSCP responders, you don't need CRL information in those.

There is actually a Mozilla discussion going on now, that I would encourage everybody who is interested in this project to join, and if you support it, let us know. I give you the link in the slides. And then along with that, there is going to be a three year maximum lifecycle in April 2015. I know a lot of people use 10 maybe five year certs right now, but that's going to shrink to a maximum of three years next year.

And this is going to permit a more quote/unquote, rapid changes in industry standards. Three years is still a long time, but at least it will be three years before we can see changes start taking affect, rather than five. And it also ensures that revalidation of that information is occurring every three years instead of some of these longer lifecycles.

So internal name deprecation as of November next year, cert CAs will be prohibited from issuing certificates that expire, or that have a name that has not been validated with the DNS. The next round of gTLDs,

unfortunately doesn't do any good for people who are new applicants, but if there is another round, the certificate issue would no longer be a problem. There wouldn't be any of these types of certs.

Right now, we are revoking all certs, 128 of the contract signing day with a new registry. So those are being phased… The ones that are being delegated are being phased out at a much quicker rate. But we have a certificate inspector tool that, again, is free on the DigiCert website. You can scan any network range, port range, and evaluate whether those internal names exist, and see if they're out there and get rid of them.

So this is just an example of the tool we have out there for you guys, you just put in a domain, it will give you a rating of what they're doing, for example, whether the certification was issued in accordance with the applicable policy, this cert is missing an AAI field, which is required. It tells you the signing algorithms being used, whether it's a SHA-1 versus SHA-2 hash, cyber-suits, things like that. So you can actually go and detect a lot of these problems, out there, so we've still seen MD 5 certs coming through, which is kind of ridiculous.

But you can see a lot of this information that I just talked about there. I think that's the end of my slides, right? No, just one. Certificate authorization, CAA. This is actually something we would like in BIND, I don't know if she's still here, but oh yes. I don't think BIND supports CAA records yet. It does? Sweet. That is awesome.

Because right now, there is a battle going on that's going to pass, that says that CAs have to at least indicate whether or not their indicating CA records. And hopefully that will become a must check in the latter,

where we actually have to look at CA records, and see which CAs are authorized. The advantage of this is it's going to reduce unintended miss-issuance of certificates because we'll know, you'll be able to specify any DNS record itself, which CAs are approved for issuance and which ones aren't.

Compliance right now is voluntary. It's not uniformly adopted amongst the CAs. And it's only a partial solution, so that's kind of a disadvantage there. So it doesn't replace the need for say, certificate transparency and other advancements in these areas. And it makes low certificate issuance if you forget to put the CA that you're doing for large companies.

We've seen sometimes people don't exactly know which CAs are always issuing for them. But right now, it's being deployed. CAs can elect not to check CAs, but they have to publically disclose that they're not checking CAs, so you'll have a list of those CAs that are and aren't. And what they do in case they encounter a CA record.

That's not in our certificate inspector software, but we're going to add that soon. Other things that you can look at, there is the heart beat bleed bug, which I was hoping to have new numbers for, but I haven't seen anything since about June, where there was still about 300, I can't remember. There are still a lot of sites compromised.

It detects crime, beast, or breach and we also update this as more happens. Like I said, it will check internal names, expiring certificate dates, missing fields and values, and checks to see if everything is in compliance with the BRs. Hopefully this helps in connection with CT to

provide greater insight on what certs are out there for its mains, and what is being used.

And then that's just the dashboard. So here is just kind of a list of tools that I wanted to provide in the slides, you can get, you know, from the materials. So SSL labs is not DigiCert, [inaudible] is not DigiCert as well. But I wanted to share as many tools as we could give you to make sure that you can check for secured on your sites and the sites of your customers.

These are our SHA-1 sunset tools inspector tools, and if you want more information where you'd like, information about what's going on in the industry, you can contact me. And I'm happy to answer any questions. I've got four minutes, I think, right?

UNIDENTIFIED SPEAKER: Any questions? Then five minutes in front of time. All right, thank you very much.

[APPLAUSE]

UNIDENTIFIED SPEAKER: As usual, we have somebody from the program committee, close the proceedings.

UNIDENTIFIED SPEAKER: I want don't want to make this long because many of us have been in this room for three days, so I think we are all tired after the very good

discussion. But I've been involved in this tech day thing [inaudible] since the beginning, and I must say, now I realize how big progress we have made, you know, because I came to the room, I tried to count the people around.

There was like 20 people, and some people coming in and coming out, so I think, and correct me if I'm wrong, but I think there was the biggest attendance ever. And also, you know, if you observe the topics, there is the most technical tech day ever. And also, we are able to invite some people from really big companies, big names, that was really fantastic. So of course there are two factors that help us, first of all, this tech day was done in cooperation with the DNS [inaudible].

And also, of course, this place is gravy because it's full of technology companies, so it was much easier to invite such speakers, and we will have hard times to make something like that in Morocco.

[Inaudible] is confident that it's going to be great. Again, let me thank you all coming here. Let me thank [inaudible] for this great effort. You know, the meeting was precisely planned and on time, so that's great, fantastic. I wish some other ICANN events would be the same. Thank you very much.

And yeah, that's basically all, but I know Keith has something to say. Before I pass mic to Keith, let me thank [inaudible] for the [inaudible].

[APPLAUSE]

KEITH: Okay, just a few very quick remarks to wrap everything up from an [inaudible] point of view. If you still have an [inaudible] name badge, we would quite appreciate it back, so if you could drop it off at the jar, at the desk, that would be great. The other thing is, obviously, some of you have been good enough to give us feedback on proceedings over the past few days, but we'd quite like to gather some data on that.

You don't want it all being about data, so we have a survey, which you get a chance to rate the workshop and the presentations. If you can go to the meeting microsite and you'll see a link for the survey, we very much appreciate you filling that in. And then finally, I would just like to thank Dr. [inaudible] and the program committee for tech day and all of the ICANN support staff for working with us to make the workshops merge together.

I know it's a lot of moving pieces during ICANN week, but I do think there is value in bringing our constituencies together, and with that, I will leave you to the rest of your ICANN meeting.

[END OF TRANSCRIPTION]