COPENHAGEN – Tech Day (Part 2)
Monday, March 13, 2017 – 13:45 to 15:00 CET
ICANN58 | Copenhagen, Denmark

EBERHARD LISSE: Okay, let's settle down and sit down and the first part after lunch is always a bit of an issue because we have incoming traffic but the floor is to Jay Daley and he will tell us about a well-researched Domain Analytics in .nz.

JAY DALEY: Thank you. My name is Jay Daley. I'm from the .nz registry, and the talk I'm giving now is about something we call Domain Analytics. It is very similar in many ways to other things you will have heard. I missed Alex's talk on DNS Magnitude earlier today but that's related and there's a company called Data Provider that does other things that are related.

Domain Analytics is – straightforwardly, it is using data science to provide insight into domain name usage and then providing that to registrars or to registrants. It's going to be a tool box of data, but initially we're concentrating on two elements, two key techniques – algorithmic popularity ranking from authoritative DNS traffic and industry coding of domains by machine learning.

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

For us this has two key outputs. The first is new functionality on the portal we provide to our registrars and the other one is a new product for registrars to sell.

We actually had the idea and started work on this in 2012 but it has taken a remarkable amount of time to get anywhere with it and only now are we close to launch.

First of all, popularity ranking.

We capture all the DNS traffic to our servers but of course, these are authoritative servers and so we need to do something clever to measure popularity that sees past caching resolvers. So we have an algorithm. Currently it uses term frequency and verse document frequency which is a way of ranking the importance of a word in a corpus of documents and we are developing that, firstly adjusting for resolver behavior and scaling by resolver size. But in order to do that we have to accurately determine resolver centricity.

We have a blog article from the data scientist on our team who is working on resolver centricity where she explains the test that she's doing and I think we have a reasonably good mechanism there so we should be able to understand that.

And then further element, we may need to compensate for varying TTL. Everybody asks, "Doesn't TTL make an enormous

difference?" And so far it looks to us as though it doesn't make much of a difference at all, and so we need to do more tests to understand that.

This is something we did in March, 2015. These are the top domains using our Term Frequency and Verse Document Frequency thing within New Zealand and to us they look roughly correct. There in the top 5 or top 10 there are newspapers, there is the main job site in New Zealand, there are some people who are running advertising campaigns at that time. This looks reasonably correct for us.

We also looked specifically just at banks and that again looks roughly correct to us based on the size of these banks and the number of customers they have.

Industry coding then – we use something called the Australia/New Zealand Standard Industry Code. This is very, very, similar to the European industry coding system. Most of the world uses a similar coding system except the U.S. which uses one that is many years older and far less refined. And so it has 19 broad divisions, 96 subdivisions, and then right the way down to the classes.

We started off by buying two hand curated marketing databases, simply the domain name plus the ANZIC with 106,000 in one and

15,000 in the other, and used half of those as a training set for machine learning and the other half then as the testing point.

We also – which I haven't noted on the slide – look at not-for-profits with a separate classification system, and to do that we employed a student to come in and hand curate 3,000 domains for us in the not-for-profit sector because that's a simple classification scheme.

This is now at a stable point. For many of the things here, and if you want to know more about the technology underlying it you can ask Sebastian Castro who I think many of you will know and he can provide details. But we're still looking improvements. We're probably doing much more hand classification than we expected to do in order to get good with this and to get to deeper and deeper levels of the classification. I don't have an example here, but the output of the classifier will generally be in the 80+% range and then the next thing that it thinks might also be will be sub 1%, and so the classifiers when they work are reasonably certain about the answer even if the answer isn't entirely correct at times.

This provides some value for registrars. It gives them market penetration by business verticals. They may be accidentally specializing in certain business verticals. They may choose to

attempt to repeat the business in the same vertical, to sell in one particular market.

The reason for using the Standard Industry Coding is that our local statistics government body within New Zealand calculates the number of companies and the market size of those companies using that Industry Coding, and so we can show to our registrars these are the proportion of those companies that we believe have a .nz domain name and we can also almost attribute a monetary value to that as well. If there are 5,000 companies and the market is worth $5 billion and only 1,000 of those have domain names, you can start to make comparisons between different industries. They might understand that a particular industry vertical has been fully mined. There is no need to try to sell any more, or there may be some profitable ones left they haven't considered.

They can also begin to understand whether verticals, some of them, do lawyers buy more domain names than accountants, for example. Which one should you be working with?

This is an entirely made-up chart which is why it's nice pink colors, but this is the type of things that we can show a registrar. We can show them the industry breakdown on the right and we can show them their breakdown on the left so that they can understand what they're doing with that. We can show them

something along these lines where we show them the penetration by the registry and then the split of classifications across the registry and then their split themselves in a different way.

This is the split that we can show registrars as well. The numbers aren't meant to add up. Everything I'm showing you in this is illustrative. None of it's from the live systems in any way. Actually there was one slide that is, but nearly all of this is illustrative.

I was in the restaurant last night with Eberhard and he reminded me I hadn't written a presentation yet so I wrote it this morning but we have all of the documentation very nicely together so it's just run straight off and I've done this presentation to our registrars several times as well.

This is the real value then for registrants. Imagine you are, say, a flower shop owner and you advertise in the lead-up to Mother's Day and you get a 50% rise in online traffic, a 10% rise in sales. That's great. Now you might actually mistake correlation for causation here. You may think that because you've spent more money on advertising and because traffic went up, there is a causal link there, but there may not be a causal link. It may simply be a correlation that the entire industry saw the traffic go up at that point and that may simply be because it was Mother's

Day and lots of people want to buy flowers for their mother on that time. And so the only way that you can understand [whether] there is real causation there and not correlation is by having industry figures. Google Analytics doesn't provide that to you. Alexa did for a period but very crudely. And so Domain Analytics is an attempt to provide that particular value to you. It's to tackle that particular problem. What we're doing is crude but it's going to be refined over a period of time to get better at doing this.

This is the shots from the product. So people classify domains and people can see how their domain is classified. They can see alternate classifications that our machine learning classifiers find and they can change that or they can set their own reclassification. This is complicated and this uses entirely fake data and so the data may be putting you off. There is no resource record "ABBA" for example. Apologies for that.

The way our popularity score works, it works on a tuple of the Qname plus the resource record. And with that tuple together, that's what we provide a popularity score for, and the popularity score is additive so you can add together the popularity score of a number of different tuples to get an overall popularity score.

And so what we do is we have a set of automatic bundling of these tuples into services. So where there is a blank Qname – the

host portion of the Qname is blank – and/or it is www and the RR is A or quad-A, then we bundle those four different tuples and call them your main website. Where we are looking at mail-dot and an MX record or mail-dot and an A record or SMTP and MX and those things we bundle that together and call that your primary mail server. And we have a series of these things that we observe from our traffic.

For example, in the top 100 sets of tuples that we see, we regularly see Minecraft servers appear which is a marvelous one. And there are lots of VOIP servers when you look at these pairs of resource records.

So we automatically create these sets of services. The registrant can amend them themselves but we automatically create them. And we then classify them as to [whether] they're application ones or whether they're mail or whether they're web. And that's what we then build these services for comparative purposes.

This is then for a particular service, say the main website, showing the domain traffic score. This is what somebody would see. It's not meant to say "rank" on the left. Sorry. That's a mistake. It should say "score." It is an arbitrary figure. It doesn't necessarily mean anything. It's an integer number between zero and one million. That's all it is. But the purpose of it is comparative. So I can compare the blog in the category of web

against all others at business level one manufacturing and I can see a rank for it now. And it's that rank that becomes important.

I can then see how the rank has changed over a period of time as well so the rank has gone up as I have done certain things such as digital advertising or others. And I can then see what the overall set looks like so I can see my rank is out to 140 or something based on that score and to get to rank number one I would have to have a score of 95,000 or something like that. It enables me to see what I need to do in order to push my rank up.

That's the popularity product. We want to sell that effectively to registrants, but registrants need some way into our product and so we provide them a health check data in-site for free for them to be able to use – a [freemium] model. And to do that we span the DNS of all of our domains every month using the standard tools that we all have available. The only thing I think we've done is create a harness to do it on such large numbers in parallel. That's the only thing we've added to that. And we already provide that data to our registrars in our registrar portal. But the registrar, because they may not be a DNS operator, often has no interest in what we do with that, in doing anything with that, and so we want to get it out to the registrant so that they can understand what they may wish to do about it.

This is our portal and this is quite small. This is how we present it in our portal. There is a chart showing the results and then down the bottom there is a table – this is fake data – one line for every domain name and then the columns are the different checks that are produced by, in this case I think Zonemaster, and then there's a dot if that particular check has failed. So we can see that. And the way we then show that in the product is with a summary and then a recommended set of actions. We show a breakdown. This is interactive in the real product so you can click on the numbers and it drops down and shows you the errors.

One of the big things that we have done is rewrite all of the error messages that come out of Zonemaster or Zonecheck – I'm not sure which one – to put them into a terminology that helps the end person – the domain name registrant – understand the implications of those. One of them would be, for example, that there is a name server listed at the parent which is not responding. We all understand what that means but they won't understand what that means and so we explain that that may lead to a resolution delay which may then affect their customer's experience. So we try to put that data there for them to be able to use.

They can then get to see the history of our scans and they can get to share a report. So they can just create a link that they can

fire off and send off to their DNS provider or their internal technician and say, “Fix this.”

As I said, this is a toolkit and we want to add things to it over time. Here is one of our nicest ideas, though we may not ever add this. What this shows around the dial is there is one segment for each industry code. The domain name in the center is scaled by its popularity and you can see the industry segment that it’s in by the pink shading there.

The other domain names are also scaled by their popularity, and their distance to the center is determined by their Levenshtein distance from the original domain name. So the closer they are, the more similar they are, or the inverse of a Levenshtein distance.

So this lets you immediately see that I have at roughly 5:00 there, there is a domain name that is as popular as mine and it is very similar but it is in an entirely different business code. Then at round about 10:00 there is another one that is less popular than mine but is in the same industry classification and so the one on the top left may be a competitor doing something, getting too close, the one in the bottom right may be someone else entirely coincidental or it may be something else. Those are the type of things we want to add to it.

And the final thing – and this is our dilemma, it's not really technical – we have two options on how we integrate with registrars. Option one is where the registrar manages the user and embeds the site and we provide an API similar to a payment gateway where they go through, they send a key, it comes back with a token, and that's the session token used for the user to get through to see their data, and we then charge a wholesale fee for that.

The other option is that we manage the user entirely through our own public site where they sign up for it, they do credit card billing directly to us, and we become the retailer, we charge a retail fee, and we then pay registrars a commission for using their customer effectively because we like to maintain that split. Our registrars are split, some wanting option one, some wanting option two, and I'm not sure the two options are compatible.

That's it for me. Any questions?

EBERHARD LISSE: Yes. What does this mean like in English?

JAY DALEY: Sorry? What?

EBERHARD LISSE: Can you explain this in an executive summary of three sentences for a person like me with an IQ of 102?

JAY DALEY: Okay.

EBERHARD LISSE: What can I use of this for .na and how?

JAY DALEY: Okay. The executive summary, as explained really clearly in earlier slides, is that we are doing some data analysis that we can use in two ways – one, for our registrars to improve their business which we would deliver to them through an existing site, our portal, and the other is to create a new product and that new product is called Domain Analytics and has a series of tools in it that we think registrants would buy. Many of us are in close to declining markets or some in declining markets. This is introducing a new product based on the real skill set that we have and the things that we find interesting as well, which is pretty high on the list, to find a new revenue stream.

So if I can go from charging a user $1.25 a month for a domain name to charging them $1.25 for the domain plus $3 a month for

this data, then I've doubled or tripled my income. That's the executive summary.

EBERHARD LISSE: Any questions? Patrick Jones? Or is this a remote question? Remote is even more acceptable.

UNIDENTIFIED MALE: The question is from John McCormack who's asking, "Have you considered or are you using deeper classifications of registrars and that some will be mass market registrars and others will be web developers that provide services to their customers. Web usage percentages on the web developer registrars may be higher."

JAY DALEY: We don't do any classification of our registrars in that way. We do some internal classification of them for marketing and management purposes but that's not related to this classification. This classification is registrants…domain names used by registrants. And it works basically off – sorry, I may have missed that out – it works off by us grabbing the text from their website and using that text for the machine learning to do the classification with. Important point I left off there.

Any other questions?

EBERHARD LISSE: Okay then. Give him a hand.

Okay. Andrew Sullivan from Dyn is going to talk to us a little bit about what happened on the 23rd of October and shortly there around. Clicker works. You have got 40 – 45 minutes with questions. You can take all the time you want.

We have all the time we need so please feel free to engage him afterwards into questions. I'm very thankful for Dyn to make Andrew available because this is something that affects not only us as clients, though we were not affected we were a client of theirs. We didn't even notice anything had happened but we all may be able to learn a little bit what can one do to prevent it or what can one do to mitigate if it happens.

Oh, yes. And I promised him that we are going to be a very friendly audience.

ANDREW SULLIVAN: Thank you. I'm Andrew for those of you who don't know me. And as you can see from this slide I have a new corporate overlord so we are now a wholly owned subsidiary of Oracle Corporation.

This all happened before Oracle Corporation took us over so, of course, none of this could possibly happen now.

I should emphasize that I didn't actually participate in any of this so many of you will know I have been for the past couple of years the Chair of the Internet Architecture Board which means that actually I haven't done any work for Dyn for two years. So only other people actually did this work and in particular Chris Baker fed me a lot of this stuff. Everything that's in this deck, if you were at [LISA] and you saw Chris and if you were at the IAB Plenary last November, all of the content that's in here is from one of those things so then you'll be bored.

With all of those disclaimers out of the way, you might have heard that Dyn had a bad day. So Dyn has a bunch of different kinds of infrastructure and one of these kinds of infrastructure is the managed DNS for fairly large web properties and enterprises and that sort of thing, so this is different from the infrastructure that operates TLD DNS systems which is why if you're a customer of Dyn and you do DNS with us and you're a TLD, you didn't' have anything actually happen to you personally. That's because they were attacking some different servers.

In any case, there are some downstream systems from Dyn that showed bad performance that day. These are various euphemisms like dependent systems or downstream systems or

whatever. But what they really are, of course, are customers and we had some attention in various media places and so on that made for a rather bad day and a lot of soul-searching meetings in which many people who had forgotten that there were technical people working at the company came in and banged on tables a lot.

So I want to explore some of these things. I'll talk a little bit about some of the things that happened. And by the way, my boss is in the room so if I'm fired at the end of this you'll know why. I want to talk a little bit about what happened and some things that I didn't witness personally and some things I did and then I want to talk a little bit about what this might mean for the Internet stuff that we're building.

Dyn has a reasonably sophisticated anycast-based system. It's mostly transit based. And I'm going to talk a little bit about that in a minute. All sites have at least two transit providers and it's carefully arranged so that there's some resilience and so on and there's supposed to be transit diversity to avoid various peering based failures and so on. So this is in 18 sites. Here are some sites. We always have to have a map, right?

There is a separate network in China. We're not talking about the network in China. That's not the relevant bit. It's all of these

other things that are in this sort of funny blue color. That's where the attacks went.

This transit emphasis that I keep making is probably not going to seem that interesting except that it turns out later to be important because this is designed to shape the path by which traffic gets to us and so we purchase this transit from these various tier one providers – whatever tier one means now – and of course that affects the way things come to our network because as their peering arrangements change that affects how the traffic gets to us.

So here is one of the kind of diagrams that you like to see about how things get to us, and you can see that depending on how this peering relationship works, you can end up coming to us via different paths or various different paths and then sometimes people have more than one path to us.

One of the things that is really important about the way attacks are happening on the Internet today is that this is mostly an economic question. Sometimes there are other kinds of issues out there. There's just politically motivated things or whatever. But there's an awful lot of economics in this.

These are not particularly new things. You can see that there used to be bots on IRC. There was a smoke jumper and that kind of stuff. And of course there were lots of ways to monetize this

sort of stuff. You used to have click fraud, there was RDP as a service, and ransomware, and so on. All of that used to happen. But what we're seeing in recent time is essentially denial of service as a kind of mechanism by which people are extracting money in various ways.

The way that this has worked in the fairly recent past, and continues to work this way but fairly recent past, was that you had these servers. You could stand things up pretty easily on the Internet. They were cheap. You could get something within the United States or within North America anyway somewhere $150 to $190. If you wanted to go outside of there you could get sometimes sort of cheaper stuff. These are usually pretty capable machines with gigabit Ethernet and all the rest of that. So this is a cheap kind of thing. And there's this nice service from Caida, the Spoofing Report, and what this does – I don't know how many of know about CarFax but this is the thing where you can find out about a used car or whatever – the same thing is true of this. You can find out whether a given provider allows spoofing. Normally, Caida's point about this is to improve the health of the Internet by reporting on all these people who permit spoofing inside their network. Well, if you want to spoof they are a very useful way to find out, "Hey, I can go into this ISPs network and I can spoof all I want." And so you get this proof that inside a given place you can do spoofing.

That's the way that a lot of these things would happen. I'm sorry, the nice thing about standards, there are so many to choose from so this slide is all messed up because I did this in one slideware and apparently we're doing this in a different slideware now. Apologies.

So the attacker wants to do some kind of high volume Denial of Service or something like that. So you buy from the provider, you check in Caida to make sure that you've got something that allows spoofing. A key thing about this, though, is that the fundamental design of the Internet is what's enabling this. You have these open systems. We've got open protocols. The protocols are designed to allow openness and to allow reuse and so on. And now we have this open reporting about what kinds of systems are going to allow spoofing and so on. And so now the attackers can use all of this stuff as part of an attack on the very infrastructure that they're doing. So they install all of these booters/stresser kind of things – here's the interface for it. You can get these things. You can now buy this as a service in case you want to do Denial of Service, you can just go out onto the Internet – and many of you will probably know – and you can rent these things for a little while.

So at the edge what happens, of course, is that the attacker announces from some ASN and because of the way that we are connected through the paths that I showed you on that diagram

earlier, some of the traffic comes directly towards us. Some of it can come through other ASNs, so it can come to us through other networks and so on. And the effect of this then is that out there at the edge where we are, we can't actually spot the path by which these attackers are coming to us, instead they're coming through other ASNs and those other ASNs, of course, are also on the path for legitimate traffic towards us. And so that's part of the problem that we're having. We don't have a peer with these people. We can't just turn them off.

So what happened in response to all of this is that the various vendors and transit providers got better and better at scrubbing. So they'll just increase the connectivity. People have been saying to me for years – I kept worrying about this two/three years ago – I said to somebody, "I'm worried about the expansion of the volumetric attacks because eventually we're going to start to have bandwidth problems," and people laughed at me and they said, "Ha ha ha. We'll never run out of bandwidth. It just keeps expanding. It's not going to be a problem." Well, here we are.

What has happened is that the transit providers got better at scrubbing these things and they all offer you these various kinds of services where things are coming in and you can identify this stuff so that one of these ASNs along the path or whatever is going to be able to spot, "This traffic is illegitimate. It's coming

from that same ASN that seems to be launching all of these attacks so we're going to scrub it away, too." And there's details to all of that and I'm sure there are people in the room who would cheerfully expound about the details of how that works but I'm not going to bore you with it.

The key point about all of this is that what you're trying to do is narrow the population or narrow the traffic that gets to your service. And this is a general pattern for any kind of service that could be subject to Denial of Service. In our case it happened to be DNS, but it's a general pattern. You've got all of the inbound traffic to you and gradually you want to narrow it down to the so-called "legitimate" traffic – whatever you mean by "legitimate" – and those are the only things that are actually supposed to reach you. Of course, the problem there is that this is a change to the paradigm of the Internet since the paradigm of the Internet was really an open network in which you were liberal in what you accepted. This is a change of that. It's an illiberality in what you're willing to accept. You're gradually willing to throw away more and more traffic because you decide it's illegitimate.

The other thing that is slightly troubling about this arrangement is that it's a sort of ride height minimum for Internet services. We're gradually pushing up how big you have to be in order to operate a service on the Internet and I think that this is

something that we need to think very hard about in two ways. First of all, as operators we have to think about this as, "Here is something that we better be aware of. We need to be big because the attackers are getting big and we're going to need to be able to respond to that."

On the other hand, as people concerned about the growth of the Internet and the health of the Internet and so on, we ought to be concerned about this because over time what's happening is that you can't launch a service on the Internet that is going to have any significant penetration without having an enormous amount of money to back it up because you've got to operate services at this volume where you can stand gigabit attacks even though your normal traffic is under 100 megabits a second. That's a fairly significant issue for the development of new services because if people have to come on and they have to come on, what are they doing? The answer that people have right now is that they're going to go into cloud providers. But of course, what that really means is that that concentrates the operator community on the Internet even more tightly than it already is concentrated.

So one of the things that we see is the preceding thing I was describing is still about 80% of all of the attack traffic that we ever see. And then there's another 20% of attack traffic. And this so-called "attack traffic" is actually mostly garbage. So you've

got broken devices that are just sending crap. They don't understand EDNS0, they do TCP retry or whatever. We've got lots and lots of lame delegations. One of the things that we have noticed very often is that people leave the delegation to us but their account has expired and they're not paying us anymore and so the parent zone continues to point to us because the registration for instance in the registry is still live and the old name servers are there but the operator of the service or whatever has gone broke or they've shut it down, they've stopped paying us. We're still getting that traffic, of course. Now we're returning an X domain. We've got to return that over and over again. And it turns out actually that the number of recursive servers on the Internet do not respect the negative TTL even now so you get a lot of these queries. That's a significant number. And we get various botnet attacks, exhaustion attacks, and that sort of thing.

So with all of that in the background – and I'm sorry if that was tedious background that you all knew – we had a warning that something was coming back in August of 2016. So we started to see this uptick in DDoS attacks and it was targeting certain customers on the Dyn platform, and we observed this in our recursive DNS traffic. So one of the nice things is that Dyn runs a small, like a really tiny, open recursive service – it's not "open" open but you sign up and so on – and that allows us to see a

certain amount of recursive traffic. And it turns out that a tiny amount of recursive traffic gives you a pretty good hint if you can also see the authoritative traffic and so we had a clue that something was happening here.

One of the things that we noticed is that – and this is mostly Chris Baker who did this – that the attacks were sort of abnormal compared to the usual attacks that we see. If you spend any time looking at attack traffic, they all have signatures and it's pretty quick you can identify what these things look like, and these ones didn't look exactly the same. They weren't using DNS amplification in the usual ways. They were being sent to known recursive resolvers so that was another thing. And there were some common patterns.

One of the ideas here – and for some reason, inside Dyn there is a bit of jargon that has developed which they call "in protocol queries" as opposed to "out of protocol" queries – and the in protocol queries or in protocol attacks, what they're really trying to do is they're saying well, they're using the DNS exactly the way it's supposed to be used but the overall pattern turns into an attack anyway. So that's what this means. I guess I didn't scrub this slide well enough.

So what happened is, these attacks would target domains that were in fact delegated on Dyn and Dyn was answering

authoritatively for them, and then they would put another label and it was sort of pseudo random, on the front of that so it was essentially the kind of thing that people do for cash busting when they're trying to measure real user monitoring or whatever. You get this random string. Well, they were doing that as an attack pattern and it was in this particular case it was consistently a 12 character, pseudo-random string and it seemed to exclude certain values – I believe XYZ was not usually in there – so this is an example of what you would see.

One of the things that was helpful here in figuring out what was going on is that there have been some people who have deployed the 0x20 tricks. Does everyone know what 0x20 is? Does anybody not know?

Okay, so very quickly – 0x20 is really a piece of jargon for the way ASCII works, that these bits are in hexadecimal they're off 20 from one another so you can just go up the ASCII table and you either get the capital or the lower case of the same character. So ASCII has this nice property that every lower case character and every upper case character has exactly one corresponding thing. And because in DNS in ASCII, the upper case and the lower case match but they're preserved, therefore as a additional source of randomness in the query in order to protect yourself against spoofing and fake responses and so on, you can randomize the case in which you send the query and you should get back the

case in the same way you sent it and that way you can know that you're talking to the right server – this is a vast oversimplification of it but I'm trying to do this very quickly – and you can look at the draft there that is mentioned on the slide and it gives you all of the background about this. I don't think this ever got advanced but a number of servers have implemented it anyway and so people use this technique, but I don't think it's been standardized.

So one of the things about this was that it gives you a clue when you start to see these 0x20 patterns, it gives you a clue that that's a real recursive server that's really doing something, it's really trying to protect something, right, so you've got a real server that's doing some work and that was the clue to us, "Hey, wait a minute. Something's going on here." TTLs were normal and all the rest of it. So Chris who is really, really, great about this kind of thing, "Hey, wait a minute. That looks funny," and then he digs in and he finds it. And this was one of the things that he spotted.

So he looks in the recursive layer and he finds this handful of infected devices that were using Dyn's recursive resolvers, and this gave him another string to start pulling on. So that was one of the clues so he could figure out what the queries per second were. And he gets these funny whisker plots. One of the things he notices is that you know the – well the first thing he notices is

that you know the distributions are really kind of funny – you get very wide distributions versus narrow ones and so on. So he starts chasing backwards to see what it is that is sending these things and it turns out they're all these sorts of servers that have just been turned up. They're very, very, simple sorts of cases. They're all these "it worked" launch pages that you get when you do a default install, so that was a pretty good sign that these were a botnet being assembled out of various hosting providers and so on.

One of the things that was weird about this was that the traffic was overwhelmingly coming from the United States and this is not a normal – I shouldn't say "overwhelmingly" but it was really pretty heavy – this is not a very normal distribution when we see attack traffic it very often is not originating in the United States and so we were trying to figure out why.

It turned out later that the source code was leaked and as you can see, there is this nice thing that's there in case the resolver doesn't work what do you do? Well it turns out what you do is you go to Google or Hurricane Electric or Verisign or Level 3, and that makes up about 24% of our aggregate traffic, so this is the reason why 35-ish% of the traffic is coming from the United States. It's not coming from the United States. It's using a resolver that is geo-located in the United States. That's why that statistic was there. But that, of course, makes our problem

harder rather than easier because we know even less about why some infected device is using, say, Google's resolver.

So Chris figured out that this was probably – I think he figured it out later – that this was probably Mirai botnet. But in any case, it was not that long. It lasted fairly short period of time. The researchers figured that there was some kind of command and control issue maybe. There were other questions about maybe it was actually device problems, that they were not that stable yet. There is a claim – so one of the things that you have to be careful about this – is that what we've got now later is the source code from the Mirai botnet, this handle, Anna-senpai, dumped the source code and the claim there was that the botnet was about 380,000 devices. The problem, of course, is that when somebody has stolen botnet source code and then dumps it on a public forum, not always the most trustworthy source in terms of numbers and so on so you have to look at that with a bit of skepticism.

So now we've got this background and so October happens. The first thing that I want to say is that I wasn't doing any operation stuff at the time but the ops people that day had a very bad day. I think it sucks when you're in ops because people come at you and every failure is public and every victory you have nobody knows about it because it worked. So I want to say that the ops

people that day did a lot of work and they had a very long day and good for them. So thank you, ops people.

One of the things that happened was we had a number of waves that came in and that was part of the difficulty that made the day bad. So a little after 10:00 UTC it looked like there were exhaustion attacks basically. They were just running the stuff as hard as possible. So they did the 12 character subdomain pattern that I mentioned earlier. And what we saw – and I'm sorry that I can't tell you the customer names here and so on but probably you saw them in the news – the attack was actually on particular pieces of the infrastructure so it seemed that one of the things that had happened was that they had mapped the way that you could get to various parts of our infrastructure because the attacks were fairly well-concentrated. In any case, these things all came together and they were all pretty well aiming at one particular zone.

That started to happen and then around 11:20 or so UTC one of the attack sources expanded and it included a larger chunk of the infrastructure. The problem at that point was that people started to wonder whether there was a lot of spoofing and, of course, because peering and transit interact as we noticed earlier, we had some difficulty analyzing exactly what was going on. So there was one ISP in Hong Kong, for instance, which appeared to be the source of some of the attack traffic, but

depending on the exact server that that attacker was going after, the traffic went to different places. So sometimes it went to Southern California. Sometimes it went to a data center in Hong Kong. And sometimes it went to a data center in Northern California. And this starts to be weird. You've got this origin and it's going to these three different places. Why is it doing that? So you have to have really, really, good, up-to-date maps of exactly how the traffic's going to flow if you're going to do things this way. This made mitigation harder because – I won't say that our maps weren't up-to-date.

This is a diagram of why this kind of thing happens. If you've got a sort of straightforward path like in the top then you don't have any trouble. You can identify exactly where the thing is coming from. But if there's more than one path to you and it all ends up coming through – in this case Jazz Telecom – there's more than one path through, you can have some difficulty in figuring that out.

Because of the earlier pattern that I mentioned, we had been doing some fingerprinting in advance and working with the team at Flashpoint, the effect of this is that we actually could do some fingerprinting and so Chris started doing that while all of this stuff was happening. What he did was he looked for the anomalous top talkers – he used NetFlow for this – and he would try to get details about the devices. And then over time, of

course, what he could do is get those IP addresses and then he would try to fingerprint as many as possible as quickly as possible. And then in the end what he had was a set of IP addresses and devices. Because of course, remember we're living in the era of NAT. There's no reason to believe that because you're getting something from some device it's the same device every time. The chances are quite good actually that there's more than one device, particularly if you've got a V4 address that's coming in and you need to be alert to that possibility because, of course, if one thing is infected in some home net, the chances are not too bad that everything else in the house is also infected and so you've got be able to identify how many devices you've got there.

Obviously all of this stuff was partly dependent on timing, so you've got a real problem because of dynamic IP address assignment you've got a real problem in keeping those things stable. So just because the address changes from this thing does not give you 100% reason to believe that you've got a new attacker. You need instead to dig through that kind of stuff. This is mostly background on how addresses change and so on and I don't think that that's news to anybody. But you have to be alert to this possibility when you're looking at your data set because you can't actually rely on one-to-one mapping. And very quickly the value of your data set goes down because a week later you

can't do any forensics on this data set anymore because the addresses are all changed. So that's another issue that you have to worry about.

With that background in place, about 12 minutes after the first wave we get another wave. And what's interesting about this is that this was a different wave of traffic and to my knowledge anyway – maybe somebody at Dyn knows – but to my knowledge we still do not know whether this was a coordinated event, whether it was just a blind accident that somebody else launched an attack at the same time, or whether in fact this was one attacker that was sending two attacks in order to confuse things. If it's the second thing – and there's reason to suppose that. There's some evidence one way and some evidence the other way – then this is a more sophisticated attack because of course they recognized that having two fronts is harder to fight than having one front and what that does is it makes your response harder.

This traffic, of course, was different. That was part of the reason that we knew that it wasn't exactly the same thing. The other thing that was interesting about this is that it was an enormous dictionary. It was just everything you could think of. There were Microsoft support phone numbers in the queries and all kinds of stuff in there, really fun. So the idea, of course, is that probably this wasn't an IOT based thing. This probably wasn't a Mirai

botnet. It was probably some other attack. But at the moment, of course, right in the heat of that moment, you don't always know all of this. All you know is, "Man, there's a lot of stuff coming in," and you're trying to react to that as quickly as possible.

So by this point, of course, everybody is in full-on panic mode. It was not a great day. And I think that one of the things that I should say is that it's attacks like these that really tell you whether you've got extremely calm people in the house. One of the people who worked at Dyn at the time is in this room and is quite calm under pressure, and those kinds of personalities are really valuable. So one of the things that I at least learn every time one of these attacks happen is look at your ops team and make sure that you've got some of those people on there who are really good at saying, "Okay, let's do this one step at a time," because you get people who start to get rattled and things can go off the rails pretty fast.

One of the fun things about being in the part of the DNS infrastructure that we are is that people use the DNS, of course, as their indirection layer for infrastructure control. They're not really having people look these names up. They're using it for infrastructure control. And what that means is that they want very, very, short TTLs because they want to be able to change

their infrastructure very quickly, and short TTLs naturally mean that your stuff disappears from cashes quickly.

The thing is that this has the advantage that you can change stuff really fast and you can reorient your infrastructure. But if your DNS systems have a bad day, it also means that you disappear from caches. And the kind of people who want to reorganize their infrastructure all of the time are also the people who end up on the news when their infrastructure goes down and so that is one of the problems that you can face here.

So, one of the things that we see over and over again – remember, the TTL and the caching in the DNS and so on wasn't just invented for convenience. Way back in the dark ages before half of you were born when the DNS was invented, the network didn't work that well. Things broke all the time. Stuff went down. And it was really handy to have this cache that was all over the Internet that mostly just worked and kept stuff up because probably the next time you ask, the authoritative server would be up fine. And so people did those things in terms of days. And now that we do them in terms of 30 seconds the resilience of this system goes down proportionately.

Another thing that's very interesting in these kinds of attacks is that Happy Eyeballs is deployed all over the Internet now which means that you get people who query both for quad-As and As at

the same time effectively doubling the amount of traffic that you have to answer. Well, that's great. So I've got short TTLs and I've got lots and lots of queries coming in. And what happens when you just don't get an answer to the query because the server is overwhelmed and it's not working very well? The Happy Eyeballs people send two more queries. And so this starts to happen and so what's nice about this attack from the attacker's point of view – not so good from our point of view – is that they are able to cause enough exhaustion of resources that clients start asking over and over again because the TTLs are all very short. This sets up a sort of retry storm which then actually amplifies the effect of the attack.

So we've got this wave one and then do some mitigations and everything calmed down. About 15:50 UTC it was a similar attack. It came in with greater volume, and importantly, it didn't ramp up the way it had before. The previous description of this was infrastructure one, two, three, they happened at different times. Second time around, they just go on all over again. And then another attack of the same sort started and it started going at a different piece of the infrastructure, and so this was later we concluded a second Mirai botnet attack.

One of the things that we really have to say is that this second wave really showed the strength of the community. When people started to hear what was going on, and while they

weren't hearing it maybe over Twitter, they were hearing it other ways – I really, really at close to the edge here, those people are going to be mad at me – the Internet operator community really started to reach out and we got a lot of support from other people including our competitors. Also [threat] Intel companies started talking to us. And this is one of the things that I think is key about forums like this and all of the operator groups and so on, make sure that your contacts are up-to-date, make sure that you've got lots of informal contacts out there and so on, because people will reach out to you. They will help you in the event of this. We all want the Internet to work and when we see this kind of attack going on, it does a lot of damage.

One of the things though that we did figure out from this is that informal methods – and I don't think that this was news – that the informal methods that have sustained the Internet this far are great and they're really important and we shouldn't let them die, but we also have potential confusion that can come from that if it's just completely informal. And you really need a sort of consistent way to operate that that makes it work for your operations environment.

There have been attempts around this. The IETF and IAB ran a workshop that was an attempt to increase some of this informal discussion and make sure that people are linking up. There's the Denial of Service Open Threats Signaling Working Group at the

IETF which is an attempt to solve some of that. There's a lot of operational communities that are trying to do this but we're still not quite there and I think it would be valuable to do this.

We do need to figure out some standard modes of sharing this data around and I don't think that that is completely working. Pcap won't do. We've got to come up with something maybe a little more compact. There've been some format wars about that, too. That's very dull. We should figure out a way to come together on it.

There are limits to growth on these attacks. The attack follows something like infectious population growth. So what you get is infection and then you start to get resilience and then you get reinfection, but the reinfection rate is limited by the recovery rate and eventually what happens is the overall system gains some immunity against this kind of attack. So the Internet does heal but we can go through some pretty serious attacks in the meantime. It looks like this is reducing once again but we should expect that a new kind of attack is probably going to come along.

Just a couple reflections on the nature of these attacks and what they mean. The first thing that we should remember is that the Internet – the internet of course, is not one big network despite what people sometimes say. It's a network of networks, and the

whole point of this was that we're going to put as much intelligence at the edge as we possibly can. The problem that we have now is that many of these devices that we're putting are not actually that intelligent. They're pretty dumb. A sensor network or even a video camera is not going to be that intelligent, and so we do want the network to avoid making a lot of the decisions. I think that that's one of the things that has caused the Internet to grow and scale the way it has. But it's also true that we don't want every sensor on the Internet to have to have a complete firewall stack on it. That's probably not going to happen, mostly because the device manufacturers don't want to do it.

We also want people to have the incentive to upgrade their stuff out at the edge, and sensor networks and so on are not very likely to do that. But the problem here, and what I like to say these days is that the Internet is under an attack of irony. What's happening is that the success of it and its very nature is what's causing the kind of attacks to be available because when too many of those smart but dumb end points are under the control of the bad guys, then you have a problem. A lot of the press about this and so on has been, "Oh, Internet of Things. Very dangerous. Very bad stuff."

Well, the Internet of Things is simply a means to this end. You could do this with other kinds of attacks. Arguably we saw this in

the late 1990s when a lot of really badly secured systems came online with cable modems and it's just that that was cable modem speed of the late 1990s and now we've got people with 10 gigabit networks and so we've got maybe a scale problem but that's about it.

The insecurity of the devices is also not the main problem. I've seen a lot of suggestions that that's where the real issue is. It's part of the problem but it is not the only main problem. The basic issue here is that you've got a network which has all of the intelligence out at the edge and we have literally no control. We have no kind of access control of any sort from the very edge of the network to any other edge of the network no matter how dumb that device is out at the edge.

I have heard people suggesting that to solve this problem, BCP38 government mandates should happen even though in this particular case BCP38 would have done absolutely nothing. I have heard people say that what we need is an Internet driver's license, that is, in order to connect you have to get approval from your government and what I don't know is whether that means people or devices. I don't know how we would enforce that. I don't know how the rules for the United States and, say, North Korea are going to be homologated. It just seems to me to be kind of a problem.

There is a suggestion – I've heard this now several times and I believe this is coming – that what we're going to get are the big important recursive resolvers are going to get preferred access to important authoritative servers. This is the submission port for DNS. This is what we did in mail. In mail you can't use Port 25 anymore on your home network. You just can't. You've got to use Port 587 because Port 25 became full of garbage because spammers and so it's too important and so we've got to shut it down and so we did. And so every node on the network is no longer equivalent. If you're a server network, then you can run Port 25 but if you're on your home network, you can't run your own mail infrastructure at home anymore. You just can't do it.

There are lots of answers here that are chipping away at the fundamental idea of a network where each network of networks can participate on an equal footing with everybody else. I don't know if there are things the IETF could do. I think, however, that there are things in general that the network community could do. We could build on the tradition of the network of networks. For instance, I don't understand why different classes of devices that are joining a network can't actually talk about what kinds of things they are so that policies could be built dynamically.

It seems to me that network scopes are a thing that we're going to have to revive. It's true that Quench is not done anymore but there are things like that that are features that maybe we should

have had. At the same time, there are proposals along these lines that scare me quite a bit. I have had somebody tell me that the DONA Foundation's Digital Object Identifiers would have solve all of this because we would have known exactly who those devices were and we could have just gone and zapped them. I'm not really sure that I want every single device in the entire world to have a unique identifier, maybe partly because I'm not sure that it's a good idea that, say, my shoes, my thermostat, and my airplane engine all are in the same identifier space. That seems to me to be different kinds of problems.

I'm also extremely worried that anything that we would do here will invite a new kind of attack, that is, everything that you can think of involves putting a control point in the network. Control point in the network is always a good target for anybody who wants to attack the network and shut it down so I think we're going to have to think very hard about that and that's part of the reason that I'm worried about things like the submission for DNS or something like that. It becomes a big fat target.

I do think, however, that we're going to have to do something about it. So 50 years ago roughly Ralph Nader published this book on "Unsafe at Any Speed," and it's mostly famous because it destroyed the Corvair in the United States, but that's not actually its biggest contribution to the automotive industry. The automotive industry in 1959 had essentially no safety

regulations for cars. There were nothing. You could buy a 1959 Cadillac with power mirrors, power seats, and no seat belts. That was the way that it worked in 1959. By 1972 none of that was true. You couldn't buy any of those things. And I think that what happened was this remark that he made, "The roots of the unsafe vehicle problem are so entrenched that the situation can be improved only by forging of new instruments of citizen action." By "new instruments of citizen action" he meant a large government department, and I think that if you are a network operator, you should think very hard about this because these people are coming and you better have an answer for them. I guess we better have an answer for them.

So that's everything that I have to say about this. I'm more than happy to take questions, keeping in mind that I may not be able to answer all of them. If I can't answer them, though, you'll all have my e-mail address, I presume – I think it's on the first slide – and I can follow up later if need be.

So with that, the floor is open. I don't know how much time is left.

EBERHARD LISSE: Thank you very much. Any questions?

One thing I'm taking with what the first gentleman comes to the microphone is we always used to say anycasting and you're done. Now you must say you must have more than one anycast provider, isn't it?

ANDREW SULLIVAN: Yeah. One of the things that is probably true is that having multiple anycast providers will help you. This was an attack across our entire anycast infrastructure and it didn't affect the other anycast infrastructures but for that piece of the infrastructure it took all of it. It doesn't hurt you to have multiple providers. Of course, there are only so many multiple providers in the world and if you think about the potential population of all of these devices out there, it doesn't seem impossible that they could attack multiple anycast providers at the same time.

I just want to point out that a lot of the attack traffic was coming from video cameras. Video cameras have the interesting property that they need to have a lot of bandwidth because they send a lot of traffic and they need to have a certain amount of processing power because they have to do video compression and all the rest of it. So basically they're pretty powerful little computers and you can have easily 20 or 25 of these in even a house and so you have this enormous potential for attack traffic and those things can open a lot of TCP connections and keep

them up. Think about, for instance, your DNS infrastructure and what happens if somebody starts opening 100,000 TCP connections a second and what does that do to you?

MAX [FRIG]: Max [Frig], Global Village. What I read in the news was that actually the attacker or the initiator of the attack was identified and it was an individual who was a disgruntled client of your customer, one who actually wanted to take a revenge on them. I take it that's not the case?

ANDREW SULLIVAN: We have heard a lot of different reports of who the exact individual was. That was certainly one of the reports that was made. I can neither confirm nor deny it. And literally I can't confirm or deny it. That's not me evading it.

MAX [FRIG]: I thought it's above your pay grade.

ANDREW SULLIVAN: No, I don't think that we actually know. I think the customer in particular would probably not share that information with us if it were true because, of course, there are potential liability issues

here as well and so people who have this kind of problem are likely to sit on it.

With that said, there are many other plausible explanations as well of exactly who did this and what it was for. The vandalism argument about this, which is essentially what that would be, is it is possible but this was a pretty good attack. This was a pretty sophisticated attack. I think that whoever it was had thought hard about what they were going to do. Attacks out of anger are often fairly destructive but fairly brief. This lasted a long time. We've got a lot of people in the office who are pretty good at blunting a lot of these attacks. We see many attacks that nobody ever knows about. So personally I am skeptical that that is the explanation but I can't confirm or deny it. Thanks.

EBERHARD LISSE: Okay. One more.

[DMITRY KOHMANYUK]: [Dmitry], ccTLD, .ua. Just a quick thing. You mentioned this, I would call it "infrastructure domains" like when people use them for [inaudible]. I'm not talking CDN use, initial low TTL. I'm thinking maybe that's more for IETF that maybe you should have another protocol for that or QS or something. Maybe same protocol but I really think that public and private use of DNS for

the infrastructure, using the same shared open bad Internet is probably not such a good idea anymore. It probably wasn't a good idea even 10 years ago. But you just let it go.

ANDREW SULLIVAN: It could be that we shouldn't do that, but the reason that people are using the DNS for this is because it's the thing that's already everywhere. So unless we've got a way to sort of tell everybody on the Internet, "Hey, you really ought to upgrade and you ought to do it promptly and do things correctly," I'm not too optimistic about the success of that. And I will point out that IPv6 has been my entire professional career a topic of how do we get people to deploy this? So I would be worried about the potential to hound people to deploy the new system when the old system provides the functionality that people seem to want.

That said, if we do things like submission for DNS, we are going to make that world whether we like it or not.

[DMITRY KOHMANYUK]: Alright. Well let's see.

EBERHARD LISSE: Okay. Thank you very much. Any remote questions? Okay.

Then let's give him a big hand. Thank you very much.

And now Maarten Bosteels from .be will explain how he put .be into the cloud.

MAARTEN BOSTEELS: Good afternoon everyone. Before I start I would like to ask – my name is Maarten Bosteels from DNS Belgium and indeed we moved our registration system for .be last month to Amazon web services. And before I start explaining why and how we did this I would like to ask to have a show of hands of who's working for a back end registry operator in here and who of you is running part of their test systems or production systems in the cloud already or literally nobody? Interesting. Who has plans to do this or is considering making a move like this?

Okay. There's at least one. Thank you.

So what did we do? So migrating to Amazon web services in itself was not such a big deal for us. I think if you would do a simple lift and shift migration it could be done in a couple of weeks or a couple of months maybe. What's more important is that we decided to rebuild our entire registration system from code, and while doing this we also took down the wall between dev and ops.

So this is what we were experiencing before the move. It's several test systems that were supposed to mimic the behavior

of the production systems but in fact because of partly manual deployments that were going on all the time, none of the systems was exactly the same as the production system. They were all very unique snowflakes actually.

As a consequence, patching would take a long time, could take several weeks, and there was a lot of fear when we were doing deployments so that means it was a lot of manual testing going on. We had difficult handovers between development and operations. And a consequence of that, of course, was that we had infrequent deployments. I think we did like five to six major releases a year. That's something we wanted to change.

And this is what our stack looked like. As you can see, at the bottom the power, co-location services, and connectivity was provided by our vendors and everything above the blue line we were doing ourselves. The blue stuff was all done by a handful of engineers. They were very good at it, but of course you need a very broad expertise area and you need a lot of broad focus to do all of this. That's why we were looking at alternatives.

Beginning of 2015 we created a new department at DNS Belgium. It was actually a merger of the development team, the operations team, and the Q&A team, and our strategy was to start focusing on the upper layers of the stack. It's more or less similar to what Erwin was talking about in the Host Presentation

this morning – focusing less on the bottom of the stack but instead [inaudible] but start focusing more on the upper layers because that's where everything we can make a difference. I think we're all really good at the bottom of the stack but if you want to make a change for our customers we have to look at the upper layer.

Another part of the strategy that we decided was to make our infrastructure completely reproducible and testable and we also wanted to build a continuous delivery pipeline to get those releases out very early and with a lot of confidence.

The last time we did a major hardware renewal was in 2011 and was a "big bang" migration. We changed almost everything. We had new hardware, new network design, new storage solution, new co-location providers. As a consequence we had a lot of vendors to manage. Another consequence was that all our hardware was becoming to an age because we did not do any gradual renewals in between. So in 2015 we were wondering, should we do another big bang migration or should we do something else? We started to wonder, do we really need to own our own hardware or could we maybe use a cloud?

In the beginning it was like the question was now and then someone said, "Let's look at the cloud," but it was not really taken seriously. There was one guy who was persistent and he

said, “We really should have a look at all the benefits,” and that’s what we started doing. So the idea was if you pull up the blue line and we outsource all of the infrastructure, we could make a lot of resources available from the system engineers. And the idea was to use those resources to build another layer in between to automate the stuff below the blue line and also automate stuff above the blue line to make a much more predictable infrastructure. This way we wanted to avoid the configuration drift that was going on between the fact that none of the test systems were actually the same as production systems. Another goal was, of course, to make patching a lot easier so that we could much faster react if there was any vulnerabilities discovered.

So we started with an initial assessment of the services of AWS and we wanted to get to know what services they offered. We started building approved concept and we also did a rather deep risk assessment, both technically and legal risk assessments. We did various performance tests and we also made an assessment of the costs.

The conclusion of this assessment was a very positive thing in fact. We could conclude that if we would do software defined everything we could really completely avoid the configuration drift. Also because the infrastructure is completely built from code, it would be automatically documented so there is no need

to document your set-up somewhere else because it's just go to the documents what you have.

An important aspect of the conclusion of the assessment was that we wanted to encrypt everything which we didn't do before. So on premises, for instance, the communication between the application servers and the database was not encrypted. That's something that we decided to do before moving to AWS, so all data in transit and all communication had to be encrypted.

Another conclusion of the assessment was that there was a lot of high ability solutions that we could just leverage from AWS instead of inventing all custom solutions ourselves which we sometimes did before. What's also very interesting what we experienced the last two years is that the services that we used from our cloud provider they keep improving for free without us, we don't even have to install new versions, they just announced literally every week they announce improvements or new features, new services, and we can start using them or we already benefit from them for free.

The cost model is very different. It's very difficult to predict your costs actually. If you just have your monthly invoices from your co-location and connectivity provider it's very predictable. But since you can scale your systems and, for instance, the development systems, the dev systems we shut them down after

office hours our developers can even launch new environments if they need more environments to work in parallel we can just launch new environments but that makes it a little bit more difficult to predict the costs, but the benefit of being able to scale up and down and make new environments is worth it, I think. In general we expect that costs will be in total will definitely be lower than before.

So this is an overview how we built this infrastructures code. So at the bottom you have CloudFormation which is a service from AWS. It allows you to define all your resources, all your infrastructure, in YAML or in JSON. We use Ansible to generate these stacks. CloudFormation is [inaudible] which means if you reapply your stack it will only change, it will only touch, the resource that [we] changed and they will not touch anything else. So that means you can just run your stack as much as you want. It will only add or change the stuff that you do. And it's literally everything in your infrastructure that is defined in these templates.

Then we use Cloudinits to pass meta information to the virtual machines, for instance the host name. And then its Puppet that, based on the host name, will install the software and do all the configuration of the virtual machine. And in between we have our rpm repo where we use snapshots of the in-house software but also upstream software, and we use snapshots of the pulp

repo so that we can exactly predict which versions will be deployed on which environment. It makes it possible to have exactly the same version of any package that we have on the production system as on the system on which we run the unit tester integration tests. All those three layers are all sitting in version control, so in the combination of those three repositories they completely define your system as a whole.

This is an overview of the environments we have. At the top left you have what we call the controller environment. It contains services which are supporting for the other environments which we have so, for instance, PuppetMaster is in there, the pulp repo is in there, and the Master NTP service is in there. Also this environment itself is also built from code completely so we can destroy it and we can rebuild it from code any time we want.

At the top right side we have another environment which we call the orchestration monitoring and test environment. It's where we run the integration tests, where we run for instance Rundeck which is a service has several jobs which developers can use to launch a new environment, destroy a new environment, stuff like that.

And then on the left bottom we have the development environments where we have several which can run in parallel. When the unit tests pass, we built packages. They are deployed

on the automated acceptance environment. When integration tests are passed, we can deploy the complete configuration infrastructure and codes to the user acceptance environment where our users can internally test and validate the system as a whole before we can deploy it to the production.

As far as high availability is concerned, all our services, all our components, are distributed over two availability zones. An availability zone is… Amazon has – I don't know by now but I think seven or eight regions worldwide where you can put your infrastructure. Ours is located in Ireland, and every region consists of two or three availability zones and the availability zone is supposed to be completely independent from the other availability zones in your region but they do have high bandwidth connectivity between them. So all our components are set up active-active and we have elastic load balancers in front of them. Another difference between the previous situation is that we have more intelligent health checks so that ELB can discard unhealthy nodes from the cluster.

Our central database is Oracle and on the left side you can see the previous situation where we had a cluster of two nodes running in one data center and we had a stand-by database in another data center. All the applications were distributed over two data centers but they were all talking to the Oracle cluster in one data center. This means that if that data center would have

an issue we would have to do manual failover to the stand-by database.

On the right side you can see the solution we use right now on Amazon. For the database we use a higher level service from Amazon which is RDS – Relational Database Service – and they have a system that they developed themselves. It's not based on Oracle Real Application Cluster or another Oracle service and they provide for all their database vendors they support and it's using synchronous replication, and they do automatic and transparent failover. We've tested this several times and the failover is done in under two minutes.

The RDS service I think it's one of the greatest time savers for us because spinning up a new database, upgrading to another version, it's going to be done in a couple of minutes. One issue that we have is that you don't have operating system access to the database so a lot of the options for replication are out of the question. That made it a little bit difficult for us to do the migration because we had over 200 gigabytes of data that needed to be transferred from the on-premises registration system to the new system on AWS.

During the course of the migration period, Amazon came out with a new service especially for doing database migrations and we started testing it of course. It looked very promising but

when we started looking at it, it was pretty immature so we decided not to use it for the migration and we used Oracle tools to do the rather basic export-import sequence.

Also interesting to note is that we scaled up the Oracle instance during the migration. That's something that it's very easy to do. You just double the number of cores and the number of RAM that we have for database. Just to speed up the migration afterwards we scaled it down again to reduce the costs because our performance tests showed that we didn't need it on a day-to-day basis. So we only paid for this more powerful server during the migration which in the total migration took two and a half hours.

So our experiences so far – the IP addresses of our servers has changed and what's more is that they will continue to change because of the way the load balancers work they can change their IPs at any time so the registrars have to be using host names. Some of them needed a little bit persuasion to do this but in the end I think after one day we saw almost all registrars had found our new EPP server. So overall we were very satisfied with the quality of service that we got, with the performance that we saw, and some of you might have heard about the S3 outage that I think was last week or the week before happened in the U.S. Luckily, we have not been affected by it because our services are running in Ireland and also we're not depending on

S3 as much as some other users do. We only use it for back-ups and we're not directly depending on it. Of course, an AWS outage is certainly not impossible but it's something we will sweat out I guess if it does happen.

So the next step that's on our road map is to have a full disaster recovery site in another region so that we can survive a full region outage from AWS. Because our complete infrastructure is built from code it would be sufficient to have just a database replicated in another region, for instance in Frankfurt, and when the time is there to switch we could just run the stack, the CloudFormation and the Puppets code to bring up an entire registration system in Frankfurt and switch the entry points and switch over to another region.

We do use the database migration service as an ongoing replication service to keep the two databases in sync, and we already have a Disaster Recovery database still in Belgium so we still have an up-to-date copy of the database at all times.

Other steps that we have on the roadmap is to fully automate the pipeline. What I described before – the migration from development to user accept to automated testing to user acceptance testing etc. – there's still some manual tests in between, some manual steps, and we want to automate them fully and create a traffic light system that when something is

green it automatically goes to the next stage. We also want to implement blue green deployments and stuff that we're thinking about is having name servers in the cloud. It will probably not be at Amazon because they don't support enhancing your own IPs so far. Maybe further in the future to implement some multiCloud strategy where we do not depend on Amazon as the only provider for infrastructure. And even more further in the future an idea is to have server-less infrastructure, servers architecture where you don't really install virtual machines but you just provide the code that needs to be run at certain events but that's a future idea.

And then I wanted to thank the team that made all of this possible, and I'm ready to take any questions if you have them.

EBERHARD LISSE: Questions please. I see Robert getting up and Jacques reaches the microphone first.

JACQUES LATOUR: I'm Jacques with .ca. Did you do a financial analysis – ROI – to see if it's cheaper to run on Amazon than yourself?

MAARTEN BOSTEELS: Yes. We did a high-level investigation of the costs. It was not the driver to do this change. Even if it would have been a little more expensive we would have gone for this route just for all the benefits that we think there are, but it turns out to be a cheaper. I don't know the exact numbers but it will be cheaper. I think the operational expenses will be more or less the same and we don't need to invest any capital anymore. So we don't need that to buy any infrastructures.

JACQUES LATOUR: Okay. Thank you.

ROBERT MARTIN-LEGENE: Robert Martin-Legène from Packet Clearinghouse. I was wondering about, you said you were encrypting data when you send it between servers in your new infrastructure. So you obviously have some concern about somebody else getting access to your data. When you choose where you host your data are you considering if your hosting provider goes bankrupt and all the assets are sold? Because in most jurisdictions, that data actually becomes property of the hosting company and will probably be sold. You said you had a back-up. I'm glad with that. My final comment, it's not so much a question, is since it costs the same more or less to run operations, do you prefer to outsource Belgian workplaces to Ireland?

MAARTEN BOSTEELS: Okay. That's a lot of questions. First of all, for the encryption we don't expect other customers to be able to reach our virtual machines on Amazon, but of course, we would like multiple layers of security so we decided it's just Best Practice to encrypt all the traffic and all data [addressed]. As to the cost, I think it's definitely cheaper to run it on Amazon but as I said, it's not the main reason to do it.

What about employment? I think we expanded our team so it's not that we want to reduce the number of ops guys or technical guys at our team but we just wanted to [inaudible] folks on more interesting stuff. And they're all very happy to do this. I think they learned a new skill set and I think it's going to be very valuable for them in the future.

EBERHARD LISSE: And Ireland is not affected by Brexit so the stuff can easily be commute to Dublin.

Two more questions and I think we'll dispense with the coffee break. We just carry on. I think we are reaching the next presentation just now anyway.

MAX [FRIG]: Max [Frig], Global Village. That's not so much a technical question but are you aware what the legal review said about having your data in the cloud of an American supplier even if it's in a data center in Ireland?

MAARTEN BOSTEELS: Yes. Our legal guy – we had, of course, a lot of conference calls with people from AWS to get all their guarantees – but the thing is it's very hard. Your existing providers, they can become an American company [over day] without you. They will not notice you anyway. I'm not saying it's not a risk but I'm saying it's hard to avoid this thing. But we had enough guarantees from Amazon that they will not move our data to the U.S. It was part of –

MAX [FRIG]: Willingly.

MAARTEN BOSTEELS: Willingly. Yes.

EBERHARD LISSE: Alright. Thank you very much. Let's give him a hand. And the next presenter will be Benno Overeinder and the coffee break would be about four minutes so we don't do it.

[END OF TRANSCRIPTION]