FRED BAKER: -- describing your [inaudible] and getting them ready for people to collaborate on them. Paul, do you want to go first?

PAUL HOFFMAN: Sure. I have some sad news, which is, at the last meeting I had said that by this meeting, where I would be is that you'd be able to set up the Test Pad, and we would be able to start talking about how to run the various tests. That is not the case. I was on schedule for that until about a week ago, when I discovered that, in fact, doing it the way that we wanted to do it in VirtualBox would not work. And then, unfortunately that came as a bit of a surprise towards the end. So, where we're at now, for those of you who are following the GitHub Repo, you've probably been seeing that I've been [inaudible], it is not ready at this point.

The main problem that we're having is that we want to emulate 13 different root servers, and what we're looking at is delays with the root server, and I think the major test that RSSAC itself was interested in is; how do resolvers do priming and re-priming? What do they do, how do they talk to the root servers when they first come up, and then also when they re-prime? So, at this moment, the test bed has its own root in it, and you can see that, that's all fine, where all of the TTLs have been cranked down from two days to 60 seconds, so the TTL is at 60 seconds, the signature time, the signature TTLs are at 60 seconds, and so on.

So, we'll be able to do tests once we can get the root servers up, but the problem that I had is that VirtualBox doesn't do real well if you have more than, say, four or five VMs, that is, it does okay, but it adds random latency. So, that obviously doesn't work for us. The original design had all 13 of the root servers in one VM, with the idea that with 13 different addresses, and the idea was that we would add a different delay to each address.

It turns out that TC, the program in Debian that allows you to add delays doesn't work by address for delays; it only works by interfaces, so we would need to have 13 interfaces on that box. VirtualBox only allows eight, but even if we wanted to subset, there is a really interesting design issue that I brought up on the TC Support mailing list, which is causing all sorts of people to discuss how networking works in Linux, which I think is fascinating, because these are all people who know it really well, and I brought up something surprising to them.

But basically, if we have one VM with 13 interfaces on it, even though we can't have all 13, well let's say we have seven or eight, the Debian kernel looks at them and says, "These are all really similar, I'm going to treat them the same, so they all get the same delay," even though the delay has been set differently for each one, so that obviously won't work. I fought with this a whole bunch, and I'm at the point now where I am saying that we cannot do this using this design; that's pretty clear.

So, what I'm going to start investigating immediately is one of two things. One would be, does this work better under a real VM system that is fully supported, such as VMware, which [inaudible] on the MAC, that's VMware Fusion; if you're on Linux, that's VMware Workstation.

They're both fairly inexpensive, they're like less than 200 or 300 bucks. And I have a copy of both of them for other work that I do at ICANN, so I'm going to test the idea of having individual VMs for each server, and try to shift under that.

Or, this will all possibly work if I change the server VM from being Debian to being FreeBSD. FreeBSD has more rock-solid networking in it than the Linux kernel; the Linux kernel has grown, the networking has grown in many different directions, whereas the FreeBSD one seems much more plain. So, it may be that that works just fine, and then we will still have a single VM with 13 addresses but using a different way of doing the delays. And I should know that within a couple of weeks.

So, that's the current status, which is, you can't use it yet. But I feel -- I shouldn't say I feel confident that we will have this useable before the meeting in Prague, the ITF meeting in Prague, because I was confident that I would have that today. But I certainly will communicate more with the mailing list as I get further, and then at that point, we are ready to start discussing how to do the actual testing. By the way, I'm sorry, I have AVM all set up that does lots and lots of different resolvers. It only does old versions of not resolver at this point, because not resolver for building any of the newer versions, it takes some really tricky matching between the not library in it, and the folks at CZ.NIC know about this, and they said, "Well, you should only be running the most current one."

And I explained why we actually want to be checking earlier ones, and so I'm actually, at this point, in the repo you will see that it only builds old ones, which work correctly, but we want to be building new ones, and so I have already scheduled time to sit down with their developers during the hack-a-thon at the ITF in Prague, so just before we're meeting. And they said that if I am sitting there with them, we should be able to beat out how to be able to build not resolvers as well.

This would be a reasonable time, and I'll start this on the mailing list, to start talking about how do we want to test priming and re-priming? Specifically, which queries do we want to send? Do we want to send queries that only get positive, do we want to look at negatives, and things like that? So, I will start that on the list hopefully within the next week, once I get over this hurdle. But that's where we are at. And, I don't know if any of you, other than Andrew, has been actually working with the Test Bed so far. If you have, I would love to hear comments. And that's what I have for us this morning.

ANDREW MCCONACHIE: Paul, this is Andrew speaking. If you have anything you can delegate to me, like if you want some testing, given the issues you're dealing with, and given what you want to do with VMware and different types of Unix, if you've got like three or four different paths you want to experiment with, you can just throw one over the fence at me and say, "Hey, go see if this works," please delegate to me, because that's what I'm here for.

PAUL HOFFMAN: Right, thanks.

ANDREW MCCONACHIE: The fact that I'm not seeing any other hands raised, to me says maybe people haven't started using this so much, which is okay, because again, it's not working at the moment, but you could at least look at the designs and such. I will push for more of that before we have our face-to-face meeting in Prague, which is in about [inaudible] and I am certainly hoping still, I'm not going to say expecting anymore, to have this fully working in the sense that as we describe different tests, we should be able to test them for the meeting in Prague. Back to you, Fred.

FRED BAKER: Okay, thank you much. Geoff, could you talk about the project that you're doing and what your plans are?

GEOFF HUSTON: Right, so I take it everyone can hear me, can they?

FRED BAKER: I can hear you.

GEOFF HUSTON: Good enough. I sent through via Joao in January an overview description of exactly what we do in terms of trying to prod the DNS into particular behaviors. The mechanics of the online ad bay system, some characteristics of the servers that we run, which are, in this case, authoritative servers, and some limitations on what this is about. Don't forget, the overall approach here is that what we do is place in the experiment, either directly by the name itself, or by the characteristics of the experiment, a particular behavior that we wish to provoke. We then see the unique domain names to many clients, millions; have those clients head towards basically four authoritative servers, and they're not Anycast, but are directed to what we believe are the closest one of those four where the authoritative server will exhibit a behavior against the visible resolver at the end of the forwarding chain, that is actually querying the authoritative server.

We have an indirect method of measuring whether the answer makes it all the way back to the client, which is based on web pitching, there are some inherent problems in that signal, the web page is not reliable, of course, so once we get back into, did the answer make it all the way back? Generally, there are some areas of inconsistency and noise, which are just impossible to reduce. However, the good news is of course, this is a massive measurement system; it's about seven, eight million a day at the moment, and it does cover almost every single part of the Internet.

I have some feedback on that report of vague nature, and I'm really not sure what other work this particular work party would like on the description of the way the measurement works that might be helpful to it. So, that's the first item. I can pause there if there's any feedback on that part of it?

No? Good. Okay, we're now looking at two items of work that are germane to this work party which are relevant. The first is one around what has been called revolver centrality, revolver clustering. Google have been kind enough in the past to actually publish the list of IP addresses used in the DNS server [inaudible] for the [inaudible] server. Other folk are not so forthcoming, and we are looking at and starting up a measurement to try and understand the extent to which users' queries head towards the big public open resolvers and measure that.

So, task one; uncovering which IP addresses are associated with which public server on the resolver side, which we tend to use RIPE ATLAS to uncover. And secondly, the task of actually doing a little experiment to provoke a user to actually query all of their configured domain names, those configured resolvers. And Joao was just setting that up now, which is an experiment that technically says, "No matter what you ask, and whether you're validating or not makes no difference, the answer to your domain name is always -- the query is always servfail."

What we found with DNSSEC is when we deliberately do unsigned or badly signed names, the servfail response actually prompts the user to exhaustively search their repertoire of configured resolvers before giving up completely, because of the inherent ambiguity in servfail, so this will allow us to extend that work, which was at the time, looking at resolver clustering for DNSSEC validation and to a growing topic of resolver clustering, per se. Any questions on that? Paul, you have your hand up?

PAUL HOFFMAN: Thanks, Geoff. I took my hand down, because I was about to speak. I am glad you are looking into that latter one, but I wonder, have you done any tests, or do we know anything about the stub resolvers, or in some cases, the browsers on how aggressively they respond or don't respond to servfail or other failures and such like that?

I'm wondering if, for example, I could believe that there could be some code in some stub resolver in majorly used [inaudible] that says, "If you get a servfail or whatever, try again, and then stop." As compared to, "Try all of them and stop." Have you looked at all at the code that you are going to be testing, basically?

GEOFF HUSTON: I'll get on to you, Warren. No, we have not looked at that code, per se. What we can do in this experiment is set up a behavior on the authoritative server, and then look at a response from the massive infrastructure and when we hand back servfail, obviously that servfail is going to filter back through recursive, forwarders, and potentially to stubs. Exactly how stubs behave is not visible to us, all that we see of course is the referred query as it travels through the infrastructure, heading towards the authoritative server.

The only thing we can do in this particular instance, Paul, is to use unique DNS labels to ensure that whatever caching would have happened, is not happening due to long history; it's happening on this query, because every query is unique. So, the answer to your question basically is; we can't look deeply down at the stub.

PAUL HOFFMAN: So, let's just say that [inaudible] found that has two, and I've got two upstream, two resolvers in Etsyresolve.com, and I hit your ad, and I do the first test, and it gets back to me, you know, that fail. You won't be able to tell -- and let's say that I have code that goes, "Okay, fall over," you won't be able to tell the difference whether I had one upstream, or two upstreams and crappy stock codes, is that correct?

GEOFF HUSTON: That's correct.

PAUL HOFFMAN: I was hoping that that was wrong and you had figured out a way to get past that, but okay.

GEOFF HUSTON: The DNS protocol is so remarkably opaque as to why queries happen, Paul, as you well know, that our directors are just foundering on the rock of DNS.

PAUL HOFFMAN: Yeah, but I had hoped that you had somehow discovered some edge-case magic, but we can leave that alone. Okay, thank you.

GEOFF HUSTON: Warren, you had your hand up, and it's gone down. I will move on unless you jump in now?

WARREN KUMARI: Yep, I am jumping in now. So, going on from what Paul said, I think as long as we know that at least one fairly well deployed sub resolver will try at least four or five or some number, and you know, [inaudible] is deployed in enough places, we should still be able to map out everything just by doing enough queries.

But anyway, that wasn't actually what I originally put my hand up for. Have you tried just asking some of the other large public resolvers? I know they don't publish them on their websites, but I think some of them might be okay to share the IPs if you just say, "Hey, could you mind telling me where you extend these queries from?"

GEOFF HUSTON: The Quad 9 folk have been happy to do that, yes, Warren. The OpenDNS folks say it's all from the same originating AS, with their Anycast distributions, so I'm having some success with that, but I kind of like the idea of using ATLAS as a way of validating that information, because ATLAS is spread, its respect is actually helpful.

WARREN KUMARI: Yeah, I agree. You know, test what's verified.

PAUL HOFFMAN: And then following along on that is, remembering that a lot of users, at least from your earlier examples, Geoff, where you've shown where you're getting queries from, a lot of them are clearly coming from wireless operators. Do you believe it would be useful or not if we somehow put out a general question to say, "The DNS operators, [inaudible] resolvers," like the wireless resolvers to also tell us, or do you think we might get enough bad information, you know, like wrong answers, where that would be worse?

GEOFF HUSTON: I kind of worry about the amount of effort versus the quality of the information we get back, Paul. I would rather, I suppose, go further down the track of going from a DNSSEC only servfail to a generic servfail, and look at that data. And Joao is telling me he is close to getting that experiment mounted.

I will move onto the second part of the work we're doing, which I think is germane again to this study. The observation is, of course, that around 65%-66% of all queries that go to root servers get answered within its domain, and if you go further down the tree in delegation, I seem to recall Duane Wessels pointing out the number got close to 94%. It doesn't matter what the number is; the observation is that an awful lot of traffic is no. We've started looking at, does no mean no in terms of setting up the experiment where the only answer is in its domain, and looking at the extent to which a single seeded query causes a number of queries back to the authoritative, and try and understand why that's happening.

The expectation, of course, naively, was if a name does not exist, one query will illicit that response, so I should get approximately one. The servers are not uniformly paced, there's all kinds of re-query time-outs, okay, 1.3, 1.4, we're kind of okay. The preliminary answer I'm getting is that no takes around 2.1, 2.2 queries before everyone is happy with a no. Happy eyeballs contributes 25% of that overload, but the remainder of that overload is actually some kind of aggressive resolver re-query together with resolver server founds, broadcasting queries to a bunch of [inaudible]. Interesting.

Then we started looking at how many answers do you need to guess; if a name exists as unique, how many queries can you get in the first 30 seconds? And the answer again, surprisingly, disregarding Happy Eyeballs, in other words, accounting for the fact that an A query is different to a [inaudible] treat them separately, is 1.7. In other words, the DNS has a huge amount of aggressive fast re-query, and that's an area again where we're trying to understand why does no amplify more than yes, and why do both no and yes amplify inside the DNS infrastructure to the extent that it looks like around 70% additional traffic comes in for each original seated query in this kind of experiment. So, there's the two areas we're working on. Again, if there are any questions, I'll happily discuss. Paul, your hand is up?

PAUL HOFFMAN: So, Geoff, you and I talked about this, I guess more than a year ago. What is the TTO you're using on the queries?

GEOFF HUSTON: We're currently using 60 seconds, and my analysis looks at the first 30 seconds of elapsed time.

PAUL HOFFMAN: Okay good, I just wanted to make sure that you weren't still viewing TTL1, which is what you had been using.

GEOFF HUSTON: No, no, no, no. No.

PAUL HOFFMAN: Yes, that would be wonderful to understand that, because I certainly in some research I did four or five years ago didn't see that; I saw a query one response for positive, so if you could find a pattern in that, that would be great. I don't know that that will be super relevant to this work party, because that doesn't really affect the root servers, although it would be interesting if you find some consistence, like in your test, you can come to a conclusion saying, "At least this kind of query consistently does this," and you can give us a range of dates that you ran those queries.

Those of us who run root servers, such as those who are in the work party, or I'm sorry, not who run work servers, who work for organizations that have root servers such as [inaudible], L-Root and Wes Hardaker for [inaudible] root, could go look and see if we see any artifacts in there. I know that we should not, because you'll be in a normal TLD, but with enough queries, we might be able to look for your names in a date range, and see whether, for example, the recursive actually ask the root once.

GEOFF HUSTON: Yeah, of course, it depends on whether we're doing two name minimization and a whole bunch of other factors there, Paul. The reason why [inaudible] I thought it might be germane to the study is actually not the overload on positives. I mean the overload on positives is there, but you know, it's the overload on negatives that I actually thought was interesting, because in some ways, we're dimensioning the root server system against NXDOMAIN responses, because that's the lingua franca of the root server, no.

And I was just wondering to what extent the volume of what they're shoveling is actually a synthetic artifact of the way the DNS infrastructure behaves as distinct from any behavior injecting queries into the DNS infrastructure, and it certainly does look like from this preliminary work that around half of the queries that come through, that given no answer, are actually synthetic queries, based on some kind of odd behavior from inside the DNS resolver infrastructure that amplifies even negative queries, when the fact that a name does not exist seems to be amplified rather than ignored.

PAUL HOFFMAN: So, a way to tie it a little bit closer to what RSSAC has asked this work party to do would be for you in your tests, even though you're trying to see what happens at your authoritative servers, is to possibly put in, in the same test, a random query to a non-existent name in the root, so that you could say, "At X time, we sent out this query and we got this funky result, hey, you root server operators, did you see something that was also for this string, at that time? And if so, did you see it multiple times?" So, it would be the same query that you were checking on, but it would be through the same string.

GEOFF HUSTON: I take that one with a certain degree of caution, Paul, if that's okay? Before I get to you, Warren, let me just quickly say that Google and their advertising standards don't directly permit to head off into non-warned domain names. And trying to set up a synthetic TLD inside this ad would generally cause the Google bots to stand up and take notice and say, "We don't like your ad anymore," so if I was to do this, I've either got to be horrendously devious, or not do it at all. It is a difficult one. Warren?

WARREN KUMARI: I was just going to say, it's an interesting idea, but of course that NSEC may have made that harder/ [inaudible]. Which actually is -- anyway.

GEOFF HUSTON: The whole reason why we started down this sort of examination of NXDOMAIN domain, Warren, is actually we are in the long run trying to understand the extent to which aggressive NSEC is being deployed. The actual --

WARREN KUMARI: [CROSSTALK], random TLDS might fail or might be bias because of the rest of NSEC and [inaudible].

PAUL HOFFMAN: But it shouldn't be, Warren, it should not be any more biased than any of the other stuff that we are seeing at the root. I mean, aggressive NSEC is obviously not very well deployed, given the amount that we are still seeing at the root. I think what Geoff is testing, which is interesting, is; what is not being caught by aggressive NSEC? Because anyone who is doing aggressive NSEC at all would also not be sending his authoritative servers' multiple questions, because the answers should already be hashed. So, I don't think it would change that.

And Geoff, I do not want you to do anything that would endanger your ad network, so this may be a task for the work party, because we were talking about how can we run parallel tests to what you're doing, and this might be a good motivator for that, is that if we're running something through not an ad network, we can do other things. So, as we get further with looking at your material, which by the way, Joao just sent Fred and I a revised copy of that just before the call. We'll be sure to get that up fairly soon. This might be a good motivator for that, where we could actually ask the root things.

GEOFF HUSTON: Warren, you have your hand up, and then I can respond to Paul?

WARREN KUMARI: Oh, sorry, forgot to put it down.

GEOFF HUSTON: What we're doing with aggressive NSEC, just for information, is Joao crafted a very cunning ad that sends one name in first, which will come back with an NXDOMAIN, it's signed, so it will have a signed NSEC, and then exactly two seconds later, which is shorter than the TTL of the zone of 60, sends in a second unique name which falls into the gap of the first. In other words, it falls into the NSEC gap that the DNS should have loaded if it was doing aggressive NSEC caching. We do not expect to see the second query if the domain system is doing aggressive NSEC caching.

Now, there's a lot of analysis yet to happen on that data, but we are getting a feed of that data now, and have done for the past four days. So, we are starting work on that, but that's more informational point at this point. The study I suppose is just trying to measure the extent to which aggressive NSEC is being used. Thanks. [AUDIO BREAK]

FRED BAKER: Okay, thank you. Where do you guys want to go next with this project? Paul, you've got some work to do in terms of virtual machines and getting stuff going. Geoff, we had talked about having a similarly, I'm not sure how to phrase it, way to set up your kind of an experiment independently, so that somebody else could do some more queries and see some more results. Is that still an objective?

PAUL HOFFMAN: So, this is Paul. I would say especially after the conversation we just had, that it should be still an objective because there may be tests that would be useful to RSSAC, that is, that would give us interesting information about how resolvers interact with the root zone that Geoff would love to run but he cannot without endangering -- which is fine, and I'm not saying Google should change, but without endangering his contract with Google, so I would say we now have a specific case where having a parallel test set up, that does not rely on Google ads, would be actually quite useful and could augment the kind of results that Geoff and Joao are getting.

GEOFF HUSTON: I have to add, it has certainly been a challenge to keep this ad running, partly because, I think, the advertising industry and online ads have been acutely sensitive to the fact that when abused, these ads become denial of service vectors, and remarkably effective ones. And the advertisers do not wish to be associated with such evilness, no one wishes to be associated with this, so when we first started doing this many years ago, there was certainly a relatively liberal attitude to what you put inside scripts and how the scripts behave, our experience with Google has certainly indicated that Google do look very, very closely at how ads behave and certainly their initial reaction is to stop anything that's outside of some relatively strict rules, and then you have to plead for your right to come back. That's just the groundwork.

I have not deeply explored other advertisers, the big ones like Facebook or some of the other placement networks. Other researchers have certainly done similar experiments, but when you start to play with domain names and play with the redirection of the user into other domain spaces, it certainly on the face of it looks like bad behavior. To what extent other ad networks filter, police, and stop it, I do not know.

Would that I had the time and materials to explore that, I don't. We have concentrated quite deliberately on simply working within the constraints of Google because we understand how it works, and we just don't have any more time to look at other alternatives. It is an option, but certainly is a field where you are very quickly painted as evil if you deviate from pretty straightforward scripting. Thanks.

PAUL HOFFMAN: This sounds like a good topic for our meeting, is how else might we be able to get inputs. If the work party sets up a parallel testing, how else might we get inputs other than the way Geoff is doing it? Are there other ad networks? Can we do it in a reliable way by putting JavaScript on [inaudible] and such? I think that that would be a good topic, certainly for the mail rooms, but also for the [inaudible]. [AUDIO BREAK]

FRED BAKER: We've come to a point where everyone stopped talking. Shall we just close the meeting and pick up then in Prague?

PAUL HOFFMAN: Hopefully, we can pick up on the mailing list before Prague, at least for my work. That is, once I have something that is testable, I will certainly let the mailing list know, but also, I have an action item even before then to start the general discussion of assuming that my test bed worked, how do we want to run the first test? How do resolvers, how does the resolver software, not the resolvers in the wild, prime and re-prime? So, those are action items for me to do on the mailing list, but I don't think we need to have another meeting before Prague, like a phone conference before Prague.

FRED BAKER: Okay, that works. Then, I'll look for you on the mailing list, and hopefully that will be quick. Hopefully, it works out. It would be nice. And -- oh, go ahead?

PAUL HOFFMAN: -- setting up the test pad.

FRED BAKER: Okay, well let me close this call out, and we'll see you guys in Prague.

[END OF TRANSCRIPTION]