COPENHAGEN - Joint Meeting: ICANN Board & Technical Experts Group (TEG)
Wednesday, March 15, 2017 - 17:00 to 18:30 CET
ICANN58 | Copenhagen, Denmark

STEVE CONTE: If anyone is looking for the universal acceptance group, it was moved from this room to B5.1. Not that we don't want you here, but if you're looking for something other than the TEG session and it's the universal acceptance, it's B5.1. And I want to play battleship now.

DAVID CONRAD: What -- oh, wow. The party can start. Jonne is here.

>> (Off microphone.)

DAVID CONRAD: Actually, we're still doing a little logistic juggling right now. Steve and Lousewies say they're on their way, so they'll be here momentarily. We're trying to juggle with some slides right now.

>> (Off microphone.)

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.

DAVID CONRAD: Yeah, we can actually start with some introductions, if people would like. For context, this is the Board versus the technical experts, the cage match. Okay. Maybe not.

This is the -- I don't know, some number of meeting of the technical experts group. This was set up to allow for technical experts to provide input to the board. We don't provide advice, we provide input. It is -- originally, it was a closed session, but we have since opened it up, and welcome, you know, anyone who is interested in geeky-related stuff to participate.

Let's see. Would -- you know, there was apparently a question of whether -- whether or not, you know, RSSAC and SSAC are invited to this. Well, (a) it's an opening meeting, (b) there might have been some confusion because we -- what meeting was that? The Marrakech? I forget which meeting, but we had to bounce -- we had to cancel the TEG to do transition-related stuff, so instead of having the TEG meeting, we decided to have the TEG/Board cocktail, and just to make things a little more entertaining, we also threw in RSSAC and SSAC, so it was the board/TEG/SSAC/RSSAC cocktail which has now become sort of become semi-tradition, and that is, TEG members and Board members are welcome to participate in the TE- -- the cocktail tonight at Ruby at 7:00ish or something.

>> 7:00.

DAVID CONRAD: 7:00, with a bus leaving at 6:45 from in front of the --

Actually, you have a microphone.

>> So, after the session, at 6:45 we have a shuttle that is a pretty large-sized shuttle at the Bella Center, west entrance, just right around the corner from here.

At 6:45, please come on Board and join us. Thank you.

DAVID CONRAD: And Steve has arrived so, as with Jonne, the party can start.

>> (Off microphone.)

DAVID CONRAD: Exactly. Would you like to say anything?

STEVE CROCKER: Sure. My apologies for coming late. I'm absolutely delighted to see so many people here. This is fantastic. David's in charge.

[Laughter]

DAVID CONRAD: Okay. So, let's start with the intros.

Marc, if you would. Your name, yeah, company, favorite color. I don't know.

MARC BLANCHET: Marc Blanchet.

JAY DALEY: I'm Jay Daley, .NZ.

DANIEL DARDAILLER: Daniel Dardailler, W3C.

LITO IBARRA: Lito Ibarra, ICANN board.

KAVEH RANJBAR: Kaveh Ranjbar, both tech and ICANN board.

LARS JOHAN-LIMAN: Lars Johan-Liman, head of the root server operations at Netnod.

GEORGE SADOWSKY: George Sadowsky, ICANN board.

RINALIA ABDUL RAHIM: Rinalia Abdul Rahim, ICANN board.

PATRIK FALTSTROM: Patrik Faltstrom, SSAC chair.

ASHWIN RANGAN: Ashwin Rangan, ICANN staff.

CHERINE CHALABY: Cherine Chalaby, ICANN board.

MARKUS KUMMER: Markus Kummer, ICANN board.

TERRY MANDERSON: Terry Manderson, ICANN staff, director of DNS engineering and area director in the IETF for the Internet area.

ALAIN DURAND: Alain Durand, ICANN staff, OCTO research.

ASHA HEMRAJANI: Asha Hemrajani, ICANN board.

PAUL VIXIE: Paul Vixie, Farsight Security, invited guest.

JEREMY RAND: Jeremy Rand, the Namecoin project.

PAUL WOUTERS: Paul Wouters, IETF liaison.

STEVE CROCKER: Steve Crocker, ICANN board.

DAVID CONRAD: David Conrad, ICANN organization.

STEVE CONTE: Steve Conte, ICANN org staff.

CATHY PETERSEN: Cathy Petersen, ICANN org staff.

WENDY PROFIT: Wendy Profit, ICANN org staff.

JONNE SOININEN: Jonne Soininen, the IETF liaison to the ICANN board.

DAN YORK: Dan York, Internet Society with a focus on DNSSEC.

SUZANNE WOOLF: Suzanne Woolf, SSAC, RSSAC, random troublemaker.

WARREN KUMARI: Warren Kumari, IETF liaison.

ED LEWIS: Ed Lewis, ICANN org, OCTO research.

ROY ARENDS: Roy Arends, ICANN OCTO research.

MATT LARSON: Matt Larson, also OCTO ICANN research.

FRANCISCO DA SILVA: Francisco da Silva from ETSI and my company is worldwide, Sweden.

HOWARD BENN: Howard Benn, also representing ETSI.

JULIE HAMMER: Julie Hammer, SSAC.

ROD RASMUSSEN: Rod Rasmussen, SSAC.

>> (saying name) from ITU-T.

ADIEL AKPLOGAN: Adiel Akplogan, ICANN org staff, technical engagement.

GREG AARON: Greg Aaron, SSAC.

MAARTEN BOTTERMAN: Maarten Botterman, ICANN board.

JAAP AKKERHUIS: Jaap Akkerhuis, SSAC and RSSAC caucus.

LOUSEWIES VAN DER LAAN: Apologies for being late. Lousewies Van der Laan, ICANN board.

JOHN CRAIN: I was hiding in the back and thought I should come up front. John Crain, ICANN organization, chief SSR officer.

DAVID CONRAD: Okay. Thank you very much.

So, the agenda is up on the screen. This is the welcome and administrivia session.

Just to reiterate what we said earlier, if you are looking for the universal acceptance steering group meeting, it has been moved to B5.1, which is just down the hallway. However, you're welcome here as well. This is, of course, the technical experts group versus the Board cage match.

Moving right along, I guess we'll start with a presentation by Jeremy Rand of the Namecoin Project talking about Namecoin. So, Jeremy, if you want to start off.

JEREMY RAND: Hi. I'm Jeremy Rand from Namecoin, so let's get started.

A full disclosure first. I'm one of the most active Namecoin developers and I'm unaware of any Namecoin developers who may disagree with anything in this talk. However, I can't speak for all the developers about all things. We're open source project that doesn't have a clear organizational structure, so just be aware of that.

This talk was prepared in collaboration with Hugo Landau.

So, the underlying motivation of Namecoin is that humans behave nondeterministically, and by extension, any system run by humans will behave nondeterministically.

And in particular, even if a system has ground rules that are supposed to be inviolable, ground rules that are enforced by humans will be inconsistently enforced.

As one example, the U.S. Constitution lays down ground rules that say torture and bulk surveillance are off limits.

Unfortunately, those ground rules are enforced by humans, and therefore, as we all know, those rules are not enforced anywhere near as deterministically as we might hope.

And human behavior in the distant future is even more nondeterministic.

For example, predicting the results of elections becomes more difficult the further in the future they are and, therefore, predicting the political climate in a country is accordingly more difficult the further into the future you go.

And the DNS is, in large part, run by humans. This poses a risk because the people involved in operating the DNS might behave nondeterministically.

Maybe your registrar makes a mistake and let's someone else change your records, or maybe the government who owns your ccTLD might get overthrown 10 years from now and the new government decides that they don't like your name and they decide to seize it, or maybe political pressure results that in the future ICANN might implement a new policy that you didn't agree to now.

And any of these could happen and this is concerning.

So, Namecoin is an experiment to find out is it possible to build something that's vaguely similar to the DNS but with as little involvement by humans as possible, and thereby create a DNS-like system that behaves more deterministically than the DNS does. And the hope here is that a system like that will hopefully be more reliable and more secure against failure modes that are caused by humans because the system is more deterministic.

So, let's look at some existing identifier systems so that we can see how they compare to Namecoin.

Manual naming at a site, things like host files, they don't have a global namespace, meaning the names are only meaningful locally, but they are safe from non-deterministic human third parties and they have human meaningful names, so that's good.

Hierarchical naming such as DNS has a global namespace but it's not safe from non-deterministic human third parties. It does have human meaningful names. This has very good usability but it is risky as a root of trust.

Content addressing like BitTorrent, where the name is the hash, has a global namespace and is safe from non-deterministic human third parties but it doesn't have human meaningful names and the content can never change.

A variant of that is the name is the public key. Things like the .ONION domains which Tor uses. These have a global namespace and are safe from non-deterministic human third parties but again they don't have human meaningful names. The content can change, though. This type of system is safe as a root of trust but it has very poor usability. The user will see a URL like you see on the screen when they try to type something in.

Actually, I'm lying. Tor is doing a security upgrade right now and when they're finished, the names will actually look like this.

[Laughter]

JEREMY RAND: Yeah. You may have noticed in the preceding slides there were two checks and one X, and this is Zooko's Triangle. So, Zooko Wilcox conjectured that it was impossible to achieve all three of these at once.

Moving on to a slightly different topic, append-only public logs are seeing increasing popularity to ensure accountability. The most successful example of this is Google's certificate transparency. Every single certificate being used on the public web is being put into an append-only log, and eventually browsers will probably require certificates to be logged to be valid. And even if you want to keep control over a system, you might want all actions to be published.

Certificate transparency is an append-only log for certificates but it's not very suitable for use with systems like the DNS, and the reason for that is who can write to the log? Anyone. But only certificates from recognized certificate authorities can be written. This is good for ensuring the logs don't get spammed with junk data, but a manual list of trusted entities is somewhat cumbersome.

Namecoin is an append-only log for name registrations and updates. However, unlike certificate transparency, Namecoin is implemented using a blockchain, so it can prevent spam by imposing an economic cost to write data, and this cost is small but very effective, and this disincentivizes bad actors from mass squatting on names without relying on a manual list of trusted entities.

Namecoin has a global namespace, is safe from non-deterministic human third parties and has human meaningful names, so it's a solution to Zooko's Triangle. Namecoin means that an append-only log for naming can be operated as an open forum, enhancing its utility. Accountability and transparency can therefore be made cryptographically verifiable public good. And independently of the system of rules that Namecoin uses for names, its nature as an append-only log means that if a bad actor does something, you always know.

As a thought experiment, consider the idea of an accountable root zone. Accountability can satisfy otherwise suspicious parties that nothing sketchy is going on.

As a hypothetical example, maintaining the root zone as an append-only log to satisfy countries worldwide that U.S. control isn't being abused even at the intergovernmental level.

Root servers could feed directly from the log. A root zone maintained as an append-only log could satisfy countries that, for example, their ccTLD won't be interfered with for political reasons, somewhat analogous to seismic monitoring used by countries to check on each other under the nuclear test ban treaty securing peace. Trust but verify. And to be clear, I'm not recommending that this particular idea be implemented in the DNS but it's an interesting hypothetical case study.

Shifting gears slightly, a related problem is the TLS public key infrastructure. The certificate authority system that is used today is problematic even with certificate transparency. And the underlying problem here is that there are way too many nondeterministic humans involved who can make mistakes. DNSSEC and DANE, which store TLS data in the DNS, rather than having certificate authorities verify them, might improve the situation. Unfortunately, there are political issues there, too. Some people are nervous about the possibility of abuse by the DNS root or the TLD operators.

And, again, the problem here is that the DNS root and the TLD operators have humans involved as well. So, it doesn't fully solve the problem of humans being involved. Namecoin could provide the advantages of DNSSEC and DANE for this purpose without the political problems.

So, we don't expect that most software or even most name resolution libraries will be aware of Namecoin directly. Instead we expect that Namecoin-to-DNS bridge software will be installed locally, translating DNS queries into Namecoin queries and converting the Namecoin responses back into DNS.

Namecoin uses the .BIT top-level domain, and this is not currently registered with ICANN or IETF right now. And we'd like to find a workable way to fix that. We realize that's a problem. For example, we might use the special use name registries, like .ONION was by Tor.

Our reference of limitation called NCDNS acts like an authoritative DNS server for the .BIT top-level domain running on localhost. DNSSEC users generate an install time, and we intentionally try to keep Namecoin's domain name specification easily mappable to DNS so that bridge software can be easily used.

If hypothetically you wanted to use this, you could tell your recursive DNS server, for example, Unbound to use NCDNS as authoritative for .BIT and supply it with NCDNS' DNSSEC public key. In theory, everything should just work. And this is only a few lines in unbound.com.

In practice, there are some DNS features that aren't very widely supported. For example, DANE for TLS. So, we have to do some weird multiplication customizations to make that stuff work. And I was actually once trying to keep track of how many different layers of crazy witchcraft we were using to make Namecoin's DANE work properly for browsers that don't support DANE for TLS. And I stopped counting at five layers of witchcraft.

So, what are some real-world use cases where Namecoin's deterministic behavior can help us? Well, let's say you are trying to buy or sell a name. In DNS, buying or selling a name usually involves some counter-party risk, and you may have to rely on an escrow agent to mitigate that counter-party risk.

In Namecoin, the buyer and seller can jointly construct a transaction that atomically pays the seller and transfers the name to the buyer. And this eliminates counter-party risk without requiring to services of an escrow agent.

And that's great, but what if the buyer and seller don't even want to talk to each other in order to set up the atomic transaction? You can buy or sell offers. And the workflow works something like this. Alice can create a sell offer. I'm willing to sell the domain name example.bit for 100 Namecoins. And Alice signs the sell offer with her private key which proves she owns example.bit and is willing to transfer it in exchange for 100 Namecoins. And Alice can post this signed sell offer on a forum or pastebin or anything like that.

Bob sees the offer and wants to buy example.bit. Bob can complete the offer by signing it with a private key that owns 100 Namecoins. And this offer is now a valid Namecoin transaction. Bob can then broadcast to the Namecoin network without contacting Alice again.

Alice gets paid. Bob receives the domain. And this transaction is atomic. There's no counter-party risk, and there's no escrow agent needed. And this works for both buy offers and sell offers. The Namecoin protocol supports this use case already, and user-friendly tools are hopefully coming soon.

In addition, another example use case is that a name is usually owned by a single private key but you can also have it owned by multiple private keys where M-of-N keys need to be present in order to issue an update. And this can be a useful protection against a single compromised key. For example, a Board of directors might each have a private key and updating the name might require a supermajority of the board. And, again, the Namecoin protocol supports this use case, and user-friendly tools are hopefully coming soon.

Namecoin can also allow very flexible update policies to be built, which can be used to customize things based on the security and UX needs of a name owner. For example, let's say Alice owns a name but she wants to limit the risk of her private key being stolen but without introducing too much counter-party risk. So, she can construct a policy that's something like this: Alice can contract Trent to run a two-factor authentication service. Alice can then update her name with arbitrary data, if Trent signs her updates. And Trent promises only to do this after verifying via two-factor authentication.

But in addition, Trent can presign specific transactions for certain events where Alice may want to do something without Trent's approval later. For example, maybe Alice wants to be able to revoke her TLSA record so if her Web server gets compromised, she can revoke the certificate easily. Or maybe Alice is concerned that Trent may disappear or go out of business or lose his private key. So, these policies can be specified based on very customizable constraints. Trent can't transfer or update date Alice's name without Alice's signature, and Alice can verify that the presigned transactions are authentic and that she is protected from Trent before she applies this policy to her name. And these policies are specified in a scripting language and are enforced to the same level that standard signatures are.

Namecoin doesn't mean that registrars go away. In Namecoin, "registrars" might look a lot like Trent. But Namecoin does mean that registrars have much less ability to harm their customers than in DNS, either accidental or malicious harm. And this might end up resulting in registrars having decreased security budgets being necessary.

Services like Trent's don't exist for Namecoin yet, but I'd like to see a service like this. As another use case, DNS infrastructure has been targeted by recent DDOS attacks, for example, the attack against Brian Krebs. And some people have suggested that Namecoin might be a useful defense. Now, it's unclear to me exactly how well Namecoin would stand up to a DDOS attack.

However, the Bitcoin network has been subjected to stress tests, which are basically DOS attack attempts in the past few years. The stress tests were conducted by for-profit companies who had a financial incentive to try to make Bitcoin's network look weak against such attacks. And Bitcoin was pretty much unaffected. Would Namecoin fare just as well? Or would attackers even have similar resources as the Bitcoin stress testers? It's hard to say. But I think it's an interesting use case. I would like to see more research on this in the future.

In order to have this determinism, however, we need to make some trade-offs. As one example, Namecoin transactions are irreversible. And as a result, if a name is transferred to a new owner, the old owner can't get it back without the new owner's signature. This means that Namecoin names are somewhat more vulnerable to hostile takeover by malware. And for that matter, human error by the name owner could also be a problem.

Some work-arounds to this would include keeping your private keys on an air-gapped machine or possibly assigning multisig or two-factor authentication policies to names, as I discussed earlier. This actually isn't all bad. I've heard security experts comment that one of the best public benefits of Bitcoin becoming popular is that people are finally taking endpoint security seriously. As Bitcoin becomes more mature, I think it is likely that endpoint security will improve substantially. So, this may be less of a problem in the future.

Another trade-off is that Namecoin doesn't have a nondeterministic human determine which name registrations are valid. And this is why it has security benefits and more resistant to political issues. However, that also means if someone registers a name that infringes on a trademark, there's no easy way to disable that name registration. You will have to negotiate with the person who registered it.

And this is pretty much inherent the definition of trademark infringement. Determining whether infringement occurred requires a human, and Namecoin is explicitly designed to not be run by humans.

A work-around for this would be users could opt into a list of known trademark-infringing names which get blocked somewhere between the Namecoin client and the user's Web browser. For example, the DNS software that is used to bridge Namecoin-to-DNS applications might support this as an option. There's already existing infrastructure for things like this. PhishTank is one example.

One caveat is a user who wants to view a name that infringes on a trademark could intentionally disable the blocking. But since the purpose of trademark law is to avoid consumer confusion, this is probably not a very big problem. A user who does this probably already knows what they're doing. Another caveat is someone could buy an infringing name solely for the purpose of selling it to the legitimate trademark owner. But since registering names cost money, it's difficult for a single person to squat on a very large number of names this way, similarly to how DNS names costing money reduces squatting.

Another trade-off is privacy. Since the full set of Namecoin transactions is public, anyone can look at the transactions. Transaction graph analysis makes it fairly easy to figure out if two transactions were done by the same person. And this also affects Bitcoin. So, what that means is, if you register two Namecoin names for different purposes, it's probably a public record that both names were registered by the same person.

And if you bought your Namecoins from someone else, they can probably see what names you registered with them. A work-around is purchasing Namecoins with a payment method that doesn't leave a public record. Meaning you shouldn't be using Bitcoins to buy Namecoins if you value your privacy. And you should also use separate public and private key pairs for each name you purchase so that they aren't linkable in the transaction graph. Bank transfers might be a good way to buy Namecoins without leaving a public record. And in addition, there were experimental efforts to make Bitcoin-like currencies that have better privacy such as Monero and Zcash that you could use to purchase Namecoins and then obtain names. They have their own drawbacks, but they may be worthwhile to some users.

And in general, the reference implementation of Namecoin has very poor privacy and makes it difficult to prevent the public from learning that all your names have common ownership. We want to make improvements on this because this is a big deal.

The last trade-off is the security of the append-only nature of Namecoin. All security properties that Namecoin has are cryptographically verifiable with one major exception, and that's that the protection of the ordering of Namecoin name operations is not cryptographically secure. Instead, it's only economically secure, meaning it would cost a lot of money to reorder the name operations. And the further back in time you go, the more money it would cost. Namecoin usually assumes that the ordering is probably immutable up to around two hours after a name operation occurs. But this isn't cryptographically guaranteed. This is probabilistic and economic in nature, so it's much weaker.

So, how could this be used for practical attack? Well, if you could reorder the transactions going back to when a name was registered, you could place a registration operation for that name before the legitimate registration, thus stealing the name.

You could also reorder the name's renewal operations to occur after the expiration period, which forces the name to expire and allows you to register it yourself. Neither of these has ever happened in real life to Namecoin. But if Namecoin gains increased adoption, more people might attempt to do it.

Bitcoin has the same problem here. But since Bitcoin's economy is much bigger than Namecoin, Bitcoin gains much more security against attacks. And there's a lot of active research into solving this issue of secondary blockchains being less secure than Bitcoin. And that's in part because a lot of improvements to Bitcoin, including some being pushed by very well-funded companies, are much easier to deploy if this issue is solved. So, we're keeping a very close eye on this research area. And we hope progress is made soon.

None of the work-arounds I just described for malware, trademarks, and privacy are quite as straightforward as the countermeasures taken with the DNS. And finding more elegant fixes is an open research problem. That said, for many real-world use cases, these work-arounds are probably sufficient.

Okay. So, where is development going? Well, unfortunately, right now Namecoin's really hard to install, especially if you want TLS support to work. And that's mainly because it's not very automated in the installation process. We just received funding from the NLNet Foundation and the Internet Hardening Fund with budget from the Netherlands Ministry of Economic Affairs. This funding will be used to improve usability and application support for Namecoin's usage as a TLS public key infrastructure. And the ultimate goal here is that Namecoin integration with a computer's name resolution system and with major Web browsers' TLS implementations will be installable in one step. So, for example, if you are on Windows, you run an .exe installer. If you are on Debian, you run a .deb package.

And this funding will also be used for UX improvements for name owners and scalability and performance improvements. And this work is being done primarily by me, Hugo Landau, Brandon Roberts and Joseph Bisch.

We're also actively engaging with the Tor project. Tor's user base has specific security requirements that are not very well-suited to the DNS. They're using .ONION now, which isn't human meaningful, and this is going to get worse when their Onion Services v3 upgrade gets rolled out as I showed earlier. And the problem is psychologically humans don't usually check the foldout onion address which means that scammers right now in the wild are creating partial pre-images of existing .ONION addresses to impersonate them. And Tor is a good candidate for early adoption of Namecoin. They can probably live with the current state of Namecoin's trade-offs, with the possible exception of the privacy issues, because all the other available options simply don't meet Tor's security requirements. And I'm the one currently leading outreach with the Tor project.

And the last area of development is on the back-end, we have an upcoming hardfork, which if you're not familiar with blockchain terminology, that's an upgrade that breaks backward-compatibility completely. And this was necessitated because Bitcoin rolled out some upgrades to their system that we can't adopt without breaking backward compatibility, and we want to stay close to Bitcoin.

We're also looking at several other upgrades, things like making the expiration period a lot more user friendly, having contact proofs of nonexistence so you can easily prove whether a name doesn't exist, allowing name point nodes to drop old data for better scalability. The hashes would still be preserved so the drop data can still be proven and also allowing Namecoins to be purchased using Bitcoins, or perhaps Monero or Zcash, without any counter-party risk. And most of these efforts are being led by Daniel Kraft.

So, thanks for inviting me. I'm happy to take any questions.

DAVID CONRAD: Okay. Thank you, Jeremy. We have a few minutes for question and answer, if anyone has any questions. Yes, Steve.

STEVE CROCKER: So, this is a great presentation. Thank you very much.

JEREMY RAND: Thank you.

STEVE CROCKER: I was paying attention about the level of protection and what kinds of things can go wrong. The strong protection is that whatever changes are made are known, as I understand it. The -- and so in the scenario for, say, root zone change, if we were adopting this, if -- if somebody changed something at the root zone, it would be known. That's a level of protection but a different problem that some parties are interested in is how can I prevent an adverse action against my top-level domain so that it just can't be done. And maybe the seeds of that are in that M of N combined with the possibility that the normal person -- the person who would normally make the change, his key will work and the other keys in concert would be used for an override or something like that. But it wasn't 100% clear to me that that's all that could happen.

JEREMY RAND: Yeah. So, yeah, you can definitely use Namecoin for the purpose of preventing malicious attacks from happening at all. Things like the multisignature method which is M of N signatures, that can definitely be beneficial for that. And similarly the example I gave with the two-factor authentication policy, that can also be used for there.

So, yeah, I think there's multiple use cases here. One use case is making sure that anything malicious that does happen is publicly known and can't be erased from memory. But yes, you're absolutely right, that it's important to be able to try to make attacks as difficult to pull off in the first place as possible. And yes, Namecoin can help with that. Since -- since the Namecoin system was originally designed for -- for end users who own a standard domain name, an idea there would be well, if you're concerned that your registrar may damage your name in some way, they may allow someone else to update it by accident, with Namecoin, if you want to, you can be your own registrar. So, you don't need to rely on a third party, unless you want -- unless you want them to be relied on for additional protection such as with multisig.

STEVE CROCKER: There's a number of cases where you may need third-party intervention, allocation of the name in the first place, recovery of keys if they've been lost, prevention or reaction to rogue behavior, et cetera. So, I have trouble envisioning a variation on the system we have that doesn't have avenues for those kinds of transactions, and, of course, as soon as you do that, then you have the exposure that you may get rogue behavior by the exceptional operator, and so it's a -- a matter of finding a good fit between those.

JEREMY RAND: Right. Yeah. So, --

STEVE CROCKER: Oh, and one more thing.

JEREMY RAND: Sure.

STEVE CROCKER: The kind of rogue operators that we're concerned about would not be concerned at all about being found out.

JEREMY RAND: Yeah, I could definitely believe that, yeah. Yeah, so yes, there's definitely a trade-off between the ability for a human to correct malicious behavior that's happened versus the ability of a -- of a legitimate user to be convinced that a human won't be able to cause damage to their own name. And yeah, this is a fundamental trade-off. There's not a very -- there's not a good way to obtain both types of protection at once. Namecoin, for this reason, is probably unlikely to completely replace DNS anytime soon. In fact, I would guess that there's a very large number of users who prefer DNS over Namecoin, for this reason. That said, there are -- I think there's also a significant user base who wants the -- the trade-offs that Namecoin makes and they're willing to put up with -- with the risk that, you know, if someone steals their private key that game -- game over. But yeah, it's definitely an open research problem into how to make protection of your private keys so good that it -- that the risk is negligible. And yeah, this is an open research problem.

STEVE CROCKER: Quick follow-up. My reading of today's technology is that stealing a private key is negligible. I mean, you put it in a bunch of hardware that if you twitch it -- but the trade-off is that you have a much higher risk that you lose control if your private key gets destroyed or lost or something like that. So, that's the action that would require recovery.

JEREMY RAND: Yeah. So, if you're not concerned about a malicious party obtaining your key but you believe that you can make sure that -- but you're primarily concerned that just your key may get destroyed by accident, then yeah. So, you can have a backup key available. You could, for example, have a multisignature policy that -- that is 1 of N. So, you could have N backups -- I'm sorry, N minus 1 backups. N1 meaning you could use the main key for everything, and you could also apply a time log so that the backup keys could only be used to recover the name if the primary key is destroyed and then, let's say, six months go by. Which isn't long enough for the name to expire but makes it, say, so that if someone tries to maliciously use one of the backup keys, they can't use it unless you've already lost the primary key as well. So, yeah, it's a fairly flexible system. But yes, at some point you are relying on that some number of keys will not get lost.

DAVID CONRAD: Okay. We have a couple more minutes for questions. Asha.

ASHA HEMRAJANI: Yeah, thank you, David. Thank you for this presentation. I have to say, I didn't quite get like maybe three quarters of it, so this is what I simplified it in my head into and I wanted to see whether I got it correct. So, one way of preventing attacks and -- so instead of the -- so instead of your domain name being under risk from, say, a government or a registrar, the DNS is in effect in your own computer, the digital phone book is sort of in your own computer.

JEREMY RAND: Yes.

ASHA HEMRAJANI: And then Bitcoin kind of ensures that every computer in the world has that same digital phone book or same DNS, is that a fair description?

JEREMY RAND: Yes. Yes, that's an excellent summary. Yes.

ASHA HEMRAJANI: Okay, cool. Whew. Okay. So, then, I want to come back to the .BIT thing that you mentioned earlier on in your slides. So, this is now referring to all .BIT Web sites, right?

JEREMY RAND: Yes. Yes. So, Namecoin is currently using the .BIT top-level domain, and as a result, so if you have the Namecoin software installed it will intercept any DNS requests for anything ending in .BIT and it will -- and it will look those up using Namecoin rather than the DNS.

ASHA HEMRAJANI: Okay. So, I have two questions. You mentioned .BIT's not registered with ICANN. Is that a requirement? For this to work.

JEREMY RAND: It's not a requirement for it to work on a technical level. I mean, it works now, even though it's not registered with ICANN. The concern is, if hypothetically in the future ICANN were to award the .BIT top-level domain to someone else, then it would not be clear how the system is supposed to work. People who have the Namecoin software installed, as it's written right now, would be accessing Namecoin Web sites using that -- using that lookup, but people who don't have it, would be accessing whatever ICANN delegated .BIT to. And people who were trying to access the other wouldn't be able to do so. And so there's a risk of namespace collision basically. And that's why we'd really like to try to get it registered officially so that there's not any risk that, you know, someone may try to buy .BIT from ICANN in the future and cause problems.

ASHA HEMRAJANI: Okay. That really helps. Thank you very much.

JEREMY RAND: Thanks.

DAVID CONRAD: Okay. Kaveh.

KAVEH RANJBAR: Thank you, Jeremy, for the presentation. I have a quick question, because to my knowledge you haven't taken this to IETF other than a bit of discussion on .BIT for the special use registry. Was it a conscious choice or are you planning to take it to IETF or not?

JEREMY RAND: That's a good question. So, when Namecoin was founded, this was back in 2011 -- and by the way, that was before I was involved in Namecoin -- the original authors had no idea that the special use names registry was a thing. And they basically figured okay, we'll just hope that ICANN doesn't -- doesn't delegate .BIT to anyone else, and, of course, this was not a very wise decision but they didn't know -- they didn't know that there was any other choice.

More recently, when three projects, Tor, I2P, and Ganu.NET, attempted to register their top-level domains through the special use names registry, we heard about it and we said, oh, that sounds like a good fit for us, too, and we contacted the authors of that Internet draft, and they added us to that Internet draft. And unfortunately, due to political reasons that I'm honestly not the best person to talk to about, that Internet draft got put on hold indefinitely. A new Internet draft did pass and become an RFC that only added .ONION, which is Tor. And so, the three other projects, GanuNET, I2P, and Namecoin, are sort of waiting for progress to be made there. But yeah, we -- we did actively engage, and maybe we weren't quite as thorough on engaging as we should have been. But yeah, once we found out that there was a process we should be following, we tried to follow that process as best we could.

KAVEH RANJBAR: Thank you very much.

JEREMY RAND: Thanks.

DAVID CONRAD: Warren and Daniel -- actually Warren and then you and closing the queue because -- for the next presentation. Warren.

WARREN KUMARI: So, one of the things which concerns me is all of your ownership of the domain is tied up in the public key -- sorry, private key, and there's a lot of sexy things you can do like M of N, et cetera, but users have a fairly hard time understanding a lot of this.

JEREMY RAND: Yes, you're right.

WARREN KUMARI: Say like with Bitcoin I can have my own private wallet and I can keep track of all my stuff myself, however, that's too complex for most people and so they use public online wallets which then get owned. Is there work going into trying to make it much simpler for users to be able to understand what exactly they're doing with this and to keep stuff local?

JEREMY RAND: Yes, there is ongoing work there. Most of that work is being done by the Bitcoin people rather than us, just because they have a lot more resources than we do. You might find the product

GreenAddress in the Bitcoin world a lot -- quite interesting. Basically, it looks like -- it's a Bitcoin wallet that you can either install as a mobile app or as a browser extension, things like that. But it has two-factor authentication under the hood. And unless you actually need to recover your keys, you know, in the event that -- that the two-factor authentication service goes down, you don't really have to worry about key management yourself, things like that. It tries to make it as user friendly as it -- as it can. And yeah, so we would really like to see systems like GreenAddress be used with Namecoin as well.

DAVID CONRAD: Okay. And Daniel.

DANIEL DARDAILLER: A couple of questions. First, you started by saying nondeterministic approach of the current DNS system was a problem, but to which extent it is a problem, you know, once you've registered your name, which goes through the registrar and the registry, then it must be deterministic. Is it the name resolver, the cache, you know, and it works like a protocol database transaction. So, what part of the deterministic, you know, problem are you trying to solve? Is it the registration itself or the resolution? That's my first question.

And then related to that, it's the question of performance. I mean, today the system is built to have, you know, very good performance because there are millions of resolutions per second, and the system using blockchain or internal append log, IP ledger, they are -- usually they have to carry the entire domain namespace to prove something using the cryptographic keys, so how does it work? I mean, considering the constraints on performance and the constraint of the append-only log.

JEREMY RAND: Yeah. Good questions. With regards to non-determinism being a problem, the example I give these days is that when the bit.ly URL shortener was registered originally, the people registering it probably didn't picture the idea that, oh, the .LY domain might be controlled by Islamic state in the future. Well, now there's a very real risk that ISIS may end up controlling that, and, you know, what happens if -- if they seize that?

In addition, domain name registrars do sometimes make mistakes. This is a lot more rare now than it used to be, but in the early days of the DNS, domain registrars have been tricked into transferring domain names to other people without proper authorization, for example, by sending forged faxes, things like that.

So, I don't think it's a very strong risk for necessarily the average case, but there's enough risk that things could go wrong that I think it's worth looking into things that behave more deterministically.

With regards to scalability, you're absolutely right that blockchains and append-only data structures in general scale much more poorly than things like the DNS, so, yeah, you're absolutely right. It's honestly not clear at this point exactly to what level something like Namecoin can scale. There was actually a fairly interesting conversation about this yesterday in the Q&A when I was on a panel here. But, yeah, it can scale quite a bit larger than it is now. I think it could handle most of the users of Tor's .ONION services without much trouble at all, which would still be quite beneficial. Could it completely replace the DNS today? Definitely not. Could it completely replace the DNS in the distant future? It's hard to say. It might, but it might not.

DAVID CONRAD: Okay. Thank you. I guess we're running a couple minutes late, so the next speaker is Paul Vixie, Farsight Security, to talk about response policy zones.

Paul, take it away.

PAUL VIXIE: Thank you, David. So, as long as we're on the topic of adding layers of witchcraft to DNS or to the naming system in general because it doesn't work the way that we want, I have my own contestant.

So, what I want to point out, though, is ICANN has gone on record, various chief executives have gone on record at various times, as saying, "We are not the Internet's police force," and this is almost invariably in response to somebody who wishes that takedown were easier, because there will be some domain name somewhere that is pointing to some resources somewhere that are causing some kind of injury to somebody and, you know, the assumption in the pre-Internet era was that everything was owned by somebody and if it was being used to harm you, you could go to -- you could figure out who that was and either get them arrested, get them law suited, or at least get them to receive your complaint and act on it.

So, this thing where the Internet is -- I don't know -- a responsibility laundering service where you keep asking for things to be taken down because they're hurting you and it turns out there's nobody who owns that and everybody says, "I'm sorry, I don't know who you could get it down to take it down but it isn't me" is very frustrating to people who are being injured by things that are happening on the Internet.

So, you know, you can complain about the weather all you want or you can get out and make some of your own.

Next slide.

So, everybody I see to my right already knows all of this, and everybody I see to my left might need a refresher, so for George Sadowsky's benefit, let me just go through this.

[Laughter]

PAUL VIXIE: There are three layers to the domain name system data flow.

At the bottom, you've got your stub resolvers. That's all your smartphones, your laptops, every VM, every M. Pretty much anything that's make a DNS query is a stub resolver. And it wants to talk to a recursive server, which frankly is not a very good name. We needed a better marketing department for this.

But forgetting what kind of recursion we're talking about, just treat it as a blank word. This thing is capable of giving you the answer to your questions, including the negative answer of there is -- there is no answer, wrong name, or no data or whatever.

It does this with a cache on the left there, so that's some storage. It's usually not disk storage as shown in the icon here, but nevertheless it remembers recent answers, so that if a lot of people ask for the same thing, that you don't have to go chasing around the Internet fetching it over and over again.

Now, if somebody asks you something that isn't in your cache, then you have to do that. You have to go up to the top level, which is where ICANN really lives. ICANN's world is the authority servers. The root name servers, the TLD servers, the effective TLD servers, the registries, the registrars, the registrants, that's all authority space.

And so, the authority servers, from a protocol point of view, are the place where content enters the domain name system from the outside. So, once it's in the domain name system, you can fetch it using the DNS protocol, but before it can be fetched, it has to be imported somehow. Normally from a text file or a database or a piece of software. And that's the job of the authorities is to import DNS content from outside.

So, what is unusual about this presentation at an ICANN meeting is that we're not going to talk about the authority servers or the policy by which you decide what name to create or who should operate whatever it is. You know, that's the -- normally, when I used to come to these things a lot more often, we spent a lot of time talking about authority server issues and policy around that, and that's certainly where all the money is, but unusually we're going to talk about the middle layer.

And the reason is that the people that are being injured using the Internet as the vector of that harm really do want to be able to do something, and it turns out you cannot stop the people who want to hurt you from registering domain names and putting content -- associating content with those domain names that will hurt you. That would be a far-end solution. If you imagine you're at one edge of the Internet and they're at the other edge, you'd like to have a far-end solution where you prevent, I don't know, brand infringement would be an example or intellectual property would be an example, child abuse materials online would be an example. There are all kinds of things that you would find harmful that you would like to be able to stop from entering the Internet at the far -- far end, but you can't, because, again, the Internet functions as a responsibility laundering service. And so, what we have evolved to, not out of choice but out of necessity, is a near-end solution, something where since I can't stop it from being created and I can't get it taken down reliably enough, I'm going to arrange my view of the Internet domain name system to be compatible with the nonexistence of whatever it is that's harming me.

And this has been very successful. We started this project in 2011. We have revised the protocol three times so we're up to Protocol 4 now. And we are currently seeking standardization of the current protocol, after which we will hand off the -- sort of the change control around the protocol to the IETF, but right now the IETF has had very little to do with this.

This really was a kind of a private team effort, not unlike the open source project like the Namecoin system, so some of you have actually contributed ideas and features to this but you did so not through the IETF but because we thought you were smart and cared what you said.

So, what we're doing here is allowing observation and analysis from the outside to be used to craft policy, and that policy then governs the response, and I'll come to the "Z" in a moment but let me just say the cache is unaffected by this.

You could imagine a policy that said, "Gee, there's a new domain generation algorithm botnet out there and it's creating all these names. It's like Conficker or whatever. And we want to make sure that if somebody looks up one of those names, they do not get an answer because the answer would -- might tell one of my bots or some infected client on my network how to reach a command and control server on someone else's network and I don't -- I have to intercept that somewhere. I choose to intercept it at DNS."

And so, you might just say, "All right, so these names, these computable names that the botnet is going to use today are forbidden," and that might be the policy you put in.

But tomorrow, that will no longer be true, right? Tomorrow has a different set of names. These domain generation algorithm botnets are using the date as part of how they compute what name to use. So, you don't want to block the name for all time. That really -- that would drastically increase the likelihood of collision, and there are collisions even though these names --

A domain generation algorithm botnet like Conficker generates really ugly-looking names but they do conflict with what I think of as ugly-looking real non-malicious names. So, you want to remove those.

And so, we do not put the policy into the cache. We actually put the truth into the cache.

So, the response policy mechanism only affects what a stub resolver will see. It does not affect what is stored or what is fetched from the authorities.

So, I told you I'd tell you about the "Z." The "Z" is the zone and reflects the fact that these recursive servers are already present in the Internet. There are 25 million of them, most of which should not be there. They are stupid little cable modems that shouldn't be running that service but do. And about 2 million of them are intentional. And so there are about 2 million recursive servers that matter. And then there's open DNS and Google with its 8.8.8.8 thing. There are a lot of recursive servers that matter. And they are -- I'll see how -- how do I want to say that?

We want to be able to control the policy of these servers using external data, and a lot of these servers are deep inside of existing networks, firewalled like crazy so that they are unable to reach the outside or to be reached by the outside.

It's considered good security hygiene to firewall your recursive name servers so they don't get used as DDOS amplifiers by people off network.

But we noticed that a lot of them are allowed to speak the DNS protocol off-net, and so we decided if we could sneak the policy in the form of a -- of DNS data to be fetched over TCP port 53 the way other DNS data is fetched, that it would probably work, and these recursive servers would be able to subscribe to a policy source. Thus, began the experiment of trying to jam response policy into the form of a DNS zone.

So, this would -- this is the ugliest DNS zone you will ever see. It is full of patterns which are intended to not occur in nature, and so it really is unnatural and it's really horrible to look at.

It's something you could be proud of how horrible it is, as though the horribleness was itself an art project.

So, the workflow here is that somebody up there on the upper right does the observation and analysis. They figure out, "Okay, a new botnet, new DGA, new set of names that shouldn't be resolved today," or maybe it's a new IP address block that you know is being used by a spammer and maybe they've got a pirate radio station and they're advertising some BGP space that isn't theirs and we really want to make sure that any answer that would result in an A record or a AAAA record that's inside of that pirated space does not get resolved today.

So, you dump all of those observation and analysis results into the response policy zone, which is then subscribed to in the normal zone transfer method by recursive servers.

Now, I wish to point out this is a voluntary act. The recursive server operator has to want this to occur. This is not SOPA. This is not something that is done to you by somebody upstream and you can't avoid it.

Furthermore, if your recursive server is subscribing to one of these things and you hate it, then you can switch to 8.8.8.8, so it's very much voluntary, even for a stub resolver. So, this whole method, although it can be used to try to effect censorship really isn't. It's -- it has to be seen as a value-add or it will not be used either by the recursive server operator or by the stub resolver operator.

So, I want to get that out also.

So, what are the numbers? A given recursive server might be running BIND or Unbound, using some software that I know of, or PowerDNS, or -- there's a fourth one. There are four independent implementations that do not share any source code with each other, and they all interoperate correctly. And in the IETF world, if you have multiple interoperable implementations, then you can start to believe that maybe the protocol document is complete enough. So, with four, I think we've got that covered.

There are thousands of recursive servers that subscribe to one or more response policy zones. And there are about a dozen security providers who publish their observation and analysis in this forum. Rod Rasmussen represents one or did until recently.

But there's a website, dnsrpz.info, that has a list of all of those implementations, all of those publishers and has pointers to the specification. And this is what the community is doing to protect itself at the near end from problems which are being introduced at the far end where we can't prevent them. And it's working. It's working really well.

We -- my company now offers a security policy in this zone format, and it has been well-received. And I think that Rod had some good success with it also in his recent company. So, it's good for the security industry because it gives us more customers for our stuff, and it's also good for people who are trying to defend themselves because it gives them a new chokepoint in their network and a very open standard that they can have a multivendor solution as to which set of security policies they want to subscribe to.

Last, but not least, this is also an enterprise solution. So, although I mentioned that Rod and I have both been in the business of selling these policies, it's also very common for, let's say, a bank to have a list of things they don't want to resolve today. And in the absence of this technology, they have been creating empty zones at any point in the namespace that they want to essentially apply a little bit of White-Out and keep things from being visible. And if you're doing 6 million of them and you're churning half of those every day, that's an awful lot of churn in your nameserver config. Whereas, on something like the response policy zone, you are not changing your nameserver config. You are just changing the response policy. It's a very lightweight operation.

So, inevitably the people who install this, the first thing they do is create a local response policy zone that is maintained by their own security department so that as they become aware of threats -- again, it's an observation and analysis again -- they can sort of quickly dump response policy into their recursive nameserver in a kind of matrix-like way where you dump it in in one place and suddenly it's synchronized everywhere and then the enterprise no longer answers certain questions or it not longer answers questions that would produce certain answers.

Other policies might be, don’t answer anything if a certain nameserver name would be involved. So, you can essentially poison content without knowing what the question or the answer is; but you know if it came from that nameserver name or a nameserver I.P. address that's in a certain range, then it's got to be bad. There's plenty of knobs. As David Conrad once told me, we have enough rope that anybody who wants to hang themselves can now do it.

And I guess the final thing to mention is mostly what we do is we say, "I want to lie" and pretend that something that does exist doesn't exist. In other words, it's a false, synthetic NXDOMAIN signal. NXDOMAIN is the return code value in DNS that indicates that the question you're asking refers to something that doesn't exist. But that's not by a long shot the only thing you can do because a lot of people don't want to do that. They would create what's called a walled garden where -- let's say that you look up a Conficker name, a Conficker botnet with a domain-generation algorithm. It might be that what you really want is to put a pop-up on your user's display to say, "Hey, you are infected with Conficker." And, indeed, you can do that if you just -- instead of answering with a synthetic NXDOMAIN, you answer with a synthetic alias to say the canonical name of what you're looking for is walledgarden.example.com. So, some Web server that's run by the enterprise itself in order to tell people, "Hey, you are probably infected, you should call the I.T. department now." So, there are a lot of other things to do besides lying about whether something exists.

And I guess truly the last topic before we get to Q&A is we are lying. These are lies. This is -- the authority is owned by somebody who you think of as malicious and you don't want the truth. And you're deciding to lie to yourself because that is the way to get your network and your assets to respond to a particular threat. And when you lie, one of the things that breaks is DNSSEC. And DNSSEC is incredibly important to the future of the world's economy. We have to have it, not just for DANE but for all the other DNSSEC-aware applications that are in the pipeline in various stages. And this breaks that. If you run this and the data itself was signed by the authority, our code will ignore it. Our code will not exercise policy over DNSSEC-signed names. And that gives bad guys a very easy way around all of this, which is they just turn on DNSSEC.

However, the stub resolver would also have to be asking for DNSSEC. So, there isn't enough DNSSEC ubiquity yet to keep this from being effective. But at some point, that's going to be a problem. And I fully expect that after we get the current specification published and turnover change control to the IETF, that there's going to be almost immediately a new protocol that is exactly like this one except it does something a little bit more sensible with DNSSEC. So, that is a known weakness that isn't affecting us now. But I really hope it does affect us because if it affects us, that means the DNSSEC became ubiquitous, which is what we need. Those are my prepared remarks, and I'm ready for Q&A. David, how many minutes have we got?

DAVID CONRAD: We probably have about five or seven minutes for questions for Paul. Who wants to start?

No questions for Paul? Okay. So, I'll start.

[Laughter]

So, Paul, I guess one implication of the RPZ stuff is that it sort of reinforces the problems that a lot of the new gTLDs are having with universal acceptance. First, is that accurate? And, two, is there some way of dealing with that?

PAUL VIXIE: So, I have a son who worked in the domain name industry for a while. And so, when .ENTERPRISES became available, he registered VIXIE.ENTERPRISES, which I thought was very cute because I had a consulting company before he was born.

And then he proceeded to try and use it and discovered that .ENTERPRISES was just not one of the patterns that, let's say United Airlines was expecting you to have associated with your account. Now, luckily, I knew the guy at United and I was able to get that fixed. But he has had all kinds of other trouble. So, I totally understand that these new generic TLDs are hard to use because a lot of people think, you know, it could be .COM, .NET, .ORG, .INFO, or a bunch of country codes. And if it isn't that, then it's got to be a syntax error. So, I get that. But that does not come from RPZ, and I have not heard of that problem in association with RPZ.

DAVID CONRAD: Okay. You indicated that RPZ doesn't work with DNSSEC. My assumption had been that RPZ worked with DNSSEC in the sense that if a zone was signed, the response came back to be validated, it could be validated. And then after the validation, the answer that was returned back to the stub resolver would be modified as appropriately indicated by RPZ.

PAUL VIXIE: That is almost true. It is certainly going to work that way if the stub is not asking for DNSSEC. If you don't set D.O. equal 1, then what you just said is what will happen. We will fetch the data. We will validate it, if possible. We will put it in the cache. And then when we are trying to firm up an answer to the original question, we will say, wait, there is policy. And the stub didn't request DNSSEC, so we're just going to -- we're going to make stuff up because if the stub is not going to be able to tell we're lying, then we will lie. However, if the stub is asking for DNS records and there are DNS records, we will not apply policy.

DAVID CONRAD: George?

GEORGE SADOWSKY: Thank you, Paul, for the refresh. I am almost ready to take the test.

So, I guess this is more a question with regard to the people who produce the information in which the policy is based.

Talk about the time-to-live considerations here. How often do you have to broadcast this? How frequently is the change that you want to give to your users? How do you know what that time-to-live is?

PAUL VIXIE: Okay. So, believe it or not, I'm glad you asked. So, the connection is live. So, the -- if you make a change, then since this is a normal DNS zone, there will be a notify and there will be incremental zone transfer, and there will be almost instantaneous updates. So, to the extent that you change your mind and you say, gee, I liked that policy ten minutes ago but I don't like it now, you can just change your mind and that will be reflected instantaneously across your subscriber base. It's very important for us that we don't break anything new. So, to that end, if -- we didn't want any stale data in the system. So, I'll give you an example.

My company sells a newly observed domain service. That's because we've observed that there are 2 1/2 new delegation points created in the Internet every second, and probably half of them will be gone in 24 hours. And 1/6 of them will be gone in ten minutes. There's a very high churn rate. These things are created for the purpose of annoying somebody, and they are taken down almost instantly in many cases or they're blacklisted by people like SpamHaus. So, that doesn't mean that everything that's new is bad, but it does mean that there's a statistical likelihood that something that is new is going to be bad.

Since I remember the good old days where you would ask for a .COM name and if it was after Tuesday, you would get it on Friday, I'm okay with new domain names not working all that well. This whole thing that ICANN and its ecosystem have developed that gets it down to 30 seconds doesn't really have a non-malicious use case that I would care about.

So, that means we have to send an update once a second to our RPZ subscribers saying, "Here are the new domain names we've observed. And, by the way, we're now deleting the ones that are more than ten minutes old because you only want the new ones and it's not new by ten minutes, by your definition." We have got different definitions.

Networks, sending an update once a second is able to synchronize the response policy across thousands of synthetic customers or dozens of actual customers. And it's all just working. So, this is very fluid. There's nothing stale.

DAVID CONRAD: Warren?

WARREN KUMARI: So, I guess this is more a comment than a question. So, I used to run my own nameserver for a bunch of domains, and then I got really annoyed with the amount of spam so I turned them off.

And then I started subscribing to RPZ feeds from a bunch of different people, and I have turned them all on again because with RPZ, I have almost no spam to deal with, right? I get spam feeds from a bunch of people with RPZ. It just takes care of the stuff and now it all works again. This is...

PAUL VIXIE: Thank you for saying so. And let me comment on your comment.

You can't get work done on the Internet unless DNS works. I know there are plenty of peer-to-peer protocols out there, and so not all BitTorrent people would notice when DNS doesn't work. But for the rest of us, if DNS doesn't work, it doesn't matter what's reachable because we're not going to be typing I.P. addresses. We are certainly not going to be typing IPv6 addresses.

Now, that property works for bad guys, too. It's not just good guys who can't get work done if DNS isn't working. Bad guys can't be reached if they're not in the DNS.

And for me, you mentioned spam. And so, to me that means email spam just off because when I was born.

I've got my mail severer, it's Postfix and it's wired up so that it tries to do a DNS lookup on every name in the header, every name in the envelope, and every name in the body. And if any of them fail, I reject the mail, which means that using this chokepoint as just a place to say these names should be non-existent, if they happen to exist, then lie and say they don't, will cause all sorts of various other failures to occur within your infrastructure. You have to be prepared for them. They can be a little surprising when you don't get that spam. In fact, what you said is a secondary intent of this whole effort.

DAVID CONRAD: Okay. Thank you, Paul, for your talk on RPZ, and now we move over to Paul Wouters.

PAUL WOUTERS: Thanks.

Since I have the microphone, a tiny comment. I have to say that Paul Vixie and John Gilmore are the two most-difficult-to-email people on the planet because of all their either defense or lack of defense mechanisms that they principally deploy.

So, with that --

>> (Off microphone.)

PAUL WOUTERS: I am happy collateral damage.

So, to get DNSSEC deployed on a large scale, that's been problematic. The DS record that people need to get into their parent zone, it's a very difficult process to get through and it involves too many humans, and the most important human is working at the registrant and he doesn't really know anything. He just bought a service and a domain name and he doesn't know anything. He just wants it to work and he has a domain operator that runs everything for him, and so they don't know even what DNSSEC is and they don't know how it enable it, and even if their DNS operator tells them what to do, they have got a really hard time of doing it.

So, there's a lot of domains that at various -- very big hosting providers that are basically signed but not delegated with a DS record, so even though they're secure by themselves, they're a little island because there's -- that DS record didn't go into the parent because there's no way of doing that.

And so, that problem needed a solution.

And the IETF first shied away from addressing it, but at some point, it became just too big a problem, so they came back and they -- they -- that's confusing. I'll just close my laptop.

So, the two things they needed to do -- and this is done now in RFC-8078 that was just published last week -- is that they need to somehow have a way for the DNS operator to signal to the registry that this domain now has a DS record in it and could you please publish this DS record.

And then the other thing that the DS operators also need to have is a way to say "My customer is moving away; my customer doesn't want DNSSEC anymore." We need some way of telling the registry to remove that record again, as well, when DNSSEC is no longer required.

And so, this RFC basically allows one to do that. What it does is it created -- it uses the record time CDS, which is basically the exact same record type as the DS but it is published at the client side, so in the client zone itself.

And so, once it's published there, you find some way of reaching your registry and say, "Hey, I've published this CDS record. Can you have a look at it and if you're okay with it, then publish it as a DS record in your parent zone."

And so, that is what this new record does.

Sorry. The record doesn't do it. The usage here of this is new.

There are various ways how you could contact your registry, and that is left to other drafts. There's currently another draft going, as an example, using a restful interface, using HTTP, to convey that information, but people could come up with other mechanisms for that as well. And then the special disable record is the CDS record with all zeros, which basically means, "Please disable this, we don't want anything."

And there's actually a typo on the slide. There should be a fourth zero, which is also an issue in the last revision of the draft, but we did catch it in time for the RFC publication but obviously, I didn't update my slide.

So, this system works. Because there's also now new EPP extensions, the registry, once they have accepted this sort of out-of-bound update from the DNS operator, they can signal this back to their registrar so that they are also aware that this record has been updated and didn't come in through a traditional EPP stream.

And this is currently being deployed or in the -- in the deployment phase for a number of TLDs, and so soon this will mean that there will be hundreds of thousands of more DNSSEC signed domains delegated, and so this should be a huge jump in DNSSEC deployment, and, yeah, we're hoping that it will be a good success.

DAVID CONRAD: Okay. Any questions for Paul?

Yes, Liman.

LARS JOHAN-LIMAN: Just a -- Lars Liman here. Just a quick clarification. This is the CDS records as was proposed by -- was it Olaf Kolkman? Or the one that's been circling in the IETF?

PAUL WOUTERS: Yeah. This is the RFC by Olaf Kolkman, I mean, yeah.

LARS JOHAN-LIMAN: All right. Thanks. And also, this kinds of puts a bit of focus on the lack of formal relationship between the DNS operators and the registries, I think, which is good.

PAUL WOUTERS: I did not mention that word on purpose.

DAVID CONRAD: Patrik?

PATRIK FALTSTROM: Have you been looking at --

To your left here.

[Laughter]

PATRIK FALTSTROM: So, what has happened in the normal case -- if you'll go back to the slide, please.

In the normal case, the DNSSEC transaction is going via the registrar, so the registrar have full state end responsibility to ensure that everything regarding the registrant is complete, including the key -- the key material.

In this case, the update of the key is going from the DNS operator to the registry without passing the registrar, okay?

PAUL WOUTERS: Correct.

PATRIK FALTSTROM: And what you are saying is that this is triggered through an event in EPP, right?

So, the registrar is supposed to use the pull command in that case to fetch the information about the new key material.

Is that the intention?

What I'm nervous about is that the registrar suddenly doesn't have a complete view of the -- of the zone, which question --

which might have an impact on the responsibilities of the registrar in relationship with the registry.

PAUL WOUTERS: That's right. Yes. But my understanding was that there was a new EPP extension that allowed the registry to push, so it's not like the registrar then need to pull.

PATRIK FALTSTROM: Absolutely. There are extensions where you can do that. But there is -- in the normal EPP design, the whole design is for registrars to update the registry and not the other direction.

PAUL WOUTERS: Correct.

PATRIK FALTSTROM: Okay. So, this is yet another thing where the registry is prescribing a change in the state machine in the registrar, and we have very, very few of those and this is another one, right?

PAUL WOUTERS: Right. However, the registrar could also support the same mechanism and then have their registrant talk to them.

So, for those who are willing to implement all of the DNSSEC requirements that don't need this work-around, they wouldn't need to make the state machine. It is -- as long as the registrar and the DNS operator have a good working relationship where they can talk to each other. Because if the registrar cannot talk to the DNS operator, then the problem remains that they cannot get this information across unless they use this mechanism.

PATRIK FALTSTROM: Can talk to you like using a DNS query, right?

Anyways, for transparency, when I reviewed this document, I suggested this document should be -- should standardize this excellent CDS record independent of whether it is the registry or registrar that do the pulling of it.

PAUL WOUTERS: Where would the registrar publish this? You mean if the registrar sends it through EPP?

PATRIK FALTSTROM: No. Publish -- the -- the registrar is fetching the new DS from the DNS operator and pushes it up to the registry using the EPP.

PAUL WOUTERS: They can already do that without this draft.

DAVID CONRAD: So, --

PATRIK FALTSTROM: Let's take this off line. Yeah. I already explained it once on the IETF mailing list and I probably don't have to do it here again.

DAVID CONRAD: Right. Dan and then Warren.

DAN YORK: So, I was just going to say thank you, Paul, for presenting this, and I think the key point, perhaps, for the ICANN Board members and the other folks who are listening here who don't want to dive into the details of some of this is just to realize that this is part of an ongoing work to provide better automation into the way that DNSSEC works, because certainly as we looked at large-scale deployments of DNSSEC by DNS operators or from other folks looking to try to do this, one of the big barriers that was identified was in getting this information, these DS records, up to the registries.

And so this is one of the mechanisms that is now available to registries that choose to make use of this to help with the automation of this publication of information and making this better which will lead to a more secure DNS in the end.

So, this is -- this is really the key point out of this is, it's a new mechanism that's available now and so registries can be looking at this as a way to make this work.

And to Patrik's point, registrars could also look at this too.

DAVID CONRAD: Warren?

WARREN KUMARI: So, the reason I started trying to interrupt Patrik is I think people were talking at cross-purposes.

I think also, Lars, you said the draft originally my Olaf. It's actually Olafur, I think, was the original -- yeah. Olafur and myself doing that. Yeah.

So, the original document didn't have the ability for people to stop publishing these records automatically. You had to go through the registry or registrar, which I think is what Patrik was talking about. We specifically left out the "you can bypass your registrar" bit because of the same concerns that Patrik was raising. This builds upon that old draft and adds new features. Or maybe I also misunderstood your --

PATRIK FALTSTROM: What I'm trying to do is just separate the technical feature, which is the ability for the DNS operator to signal that this is new key material from the potential policy impact regarding relationship between the registrant, registrar, and registry. That discussion is a completely different one that could be messy in certain TLDs.

DAVID CONRAD: Just a quick response? Yeah.

>> Patrik, the fundamental issue that needs to be solved is in figuring out who is the registrant, what is the handle to talk to it. We have a limited number of registries so they are convenient as the starting point to talk to, but if we can somehow get into RDAP or some other protocol, finding the handle for the entity willing to talk to us, preferably the registry or a reseller, even, or a reseller of a reseller, that is the thing that nobody can find today.

DAVID CONRAD: Dmitry?

DMITRY KOHMANYUK: Hi. Just want to make an quick comment why the registry box is doubled. I suppose it's a typo. Secondly, I would probably second Patrik Faltstrom's comment that, yes, the EPP model -- and I, by the way, represent one of the TLDs, ccTLD. We run EPP. It's Ukraine. I think the model when the state is split is very bad. I also think the pulling model is very bad and doesn't scale. Nevertheless -- and, yes, EPP supports DS updates, but my biggest issue here is that we are trying to separate the DNS -- sorry, NS record and DS record management. That's bad. Because changing DNS operator may involve both DS record change and DNS record change. Somehow seems strange that DS updates are supposed to (indiscernible) this draft without oversight of the DNS record.

So, I would say you should go back to the drawing Board and see how this whole separation of registrar, say, data update about entity names, addresses, and stuff versus technical data. Yes, it's a good idea to lose third-party technical operator, but the point is the wrong solution, plus lack of contractual relationship between DNS operator, one or two, and registry is wrong solution, and that's not a way to solve it and that's not a way to make the Internet more secure.

So, yeah, nice try, but I would just --

PAUL WOUTERS: So, I'll just -- a very quick note and then I'll give it to Paul Vixie.

There has been a long discussion in the IETF about triggers versus timers, and so let's not repeat that again.

It is an option that registries can decide to pick, and if some registries contractually cannot do it or don't want to do it, that's fine, but this is going to be an option that is useful for a large number of people that currently cannot push DS records where they should go.

DMITRY KOHMANYUK: Well, yeah. There are many issues. I don't think we should discuss this right here. It's better to be done in the IETF environment. Thank you.

PAUL WOUTERS: Okay.

DAVID CONRAD: Paul?

PAUL VIXIE: I was going to amplify that point. The NomCom works very hard to get the most qualified people willing to serve on this board, and they are not necessarily as technical as the people who were using the Internet in the years before ICANN existed. We must use their time wisely and respect their time, so if you could please pitch your arguments at a level that George Sadowsky can understand.

[Laughter]

DAVID CONRAD: And with that, we're now into any other business.

You know, this was an attempt to sort of restructure the way TEG operates. We provided briefings, one- to two-page briefings to the Board members prior to the meeting and was just wondering if that was beneficial or if we should continue to try to evolve the TEG in a way that makes it more useful for Board members.

And you can either say now or you can send me email or you can hunt me down at the cocktail party that is soon to follow. Buses leave in about 15 minutes. And with that, I will call this session of the TEG closed and thank you for your participation.

[Applause]

[END OF TRANSCRIPT]