Stephen Farrell – IETF Security Area Director
Interview transcript with The Register, 6 December 2016

The Register: To set the scene: if we look at how the internet was when I first connected in 1993, everybody assume the sysop could see everything. For the next 10-15 years, nobody really paid much attention to making things more private.

Why did so many protocols still arise without considering privacy and security and monitoring?

Stephen Farrell: I wouldn't quite agree with that characterisation … I would assert that people like the IAB were thinking about these topics earlier, but they weren't getting quite so much traction.

So, for example – if you look at RFC 1984 from August 1996, back in the crypto wars 1 – a lot of the technical community were concerned about these issues, but they weren't getting traction.

The reasons for that – some of those are technical. Cryptography, compared with the ability of your CPU to actually do processing, was much more constrained then than it is now.

Moore's Law has helped a lot, in terms of making things like wide-scale deployment of TLS feasible, at significant scale, whereas 10 or 15 years ago that wasn't the case. Banks, and people who did care about security back then, they had to buy TLS accelerators, special hardware boxes to put in front of their servers – today, you don't need that any more really.

So I think that's part of the reason.

I guess another part of the reason is: even though a bunch of the nerds said “you should all be worrying about this”, there's a bunch of people in the world who don't worry about stuff until something bad happens.

I think that whole … human perception of risk enters in. People don't perceive risk, not because they're stupid, but because people are not built to perceive the kinds of risks involved in networking very well. We're built to run away if an animal attacks or something, but not to consider the possibility that somebody's going to break into our router at home.

The Register: Or a server 14,000 km away.

Stephen Farrell: Yes. There are hard technical issues, some of which have been helped by Moore's Law, others which are helped by us getting better at the technology, but there are things to do with perception of risk, which is still an ongoing thing.

It's just not human nature to understand the kind of probability that's involved in these kinds of events.

Also: if you go back to the late 1990s, the Internet was nowhere near as critical for nowhere near as many of us as it is today. The importance, the ubiquity, the pervasiveness of the network is vastly different now.

The ability to do a demonstration of the downsides of not doing things, and the overall criticality of the network for our daily communications and lives means that it's more that we

actually try and do something.

And when that event happens, it means it has worse impact.

The Register: And now, we'll take a massive jump through time, and ask: Then the IAB and the IETF define a problem in RFC 7258, that starts a whole lot of conversations within the community – when that document first lands, how did the conversations go? How did people react? "We have to do all of this?"

Stephen Farrell: I can tell you at the time, I was totally sick of it, because it took about 1,000 e-mails to get that agreed! You can go back and look at them in the archives if you really care to torture yourself …

I can talk about the process: There's a whole bunch of people who wanted to get consensus that the main message of this is correct.

There were also a range of concerns from people who do various implementations, or network management, or network operations. Those concerns are real, and they're still playing out, and we're still learning about them as they go.

One of the sensible reactions – one of the mitigations to pervasive monitoring that's in use, in various ways, is encrypting things.

The more we encrypt things, the less that the network management people can see what's going on in the network; and that impacts on what they're doing.

There's two aspects [to this]: Network management people have been used to managing cleartext networks. They were used, for more than 20 years, to being able to look into packets, see what they contained, and take action on them.

Not for nefarious reasons – in order to detect attacks, in order to optimise traffic, and so on.

We're changing that, and that also means the technology they're using will be undergoing change, to deal with much more ciphertext than plaintext.

So that was a concern for a bunch of people in network management – it still is, and we need to learn better ways of how to fulfil those same functions on the network.

If you had some security mechanism in your network for detecting some malware attack traffic, instead of being able to operate that from the middle of the network, it pushes a requirement on you to move that to the edge.

If you look at some of the commercial instant messaging providers, that have introduced end-to-end encryption of their messaging – they have found they can move those functions in their networks to new places to do what they need to do.

It means change, but it doesn't make network management impossible.

But every change has somebody who would rather not.

“This is going to change my network – what am I going to do?” is a valid concern, and one we learn about as we go.

The Register: UTA, DPRIVE and TCPINC – where have we gone in those working groups, in terms of standards delivered, that people can sit down and implement?

Stephen Farrell: These are the working groups that were directly established as a result of RFC 7258 – I wouldn't say they're the most important things that happened. Things like HTTP2 and QUIC are maybe more important – I'll come back to that.

The UTA working group produced RFC 7525 (*Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)*, https://tools.ietf.org/html/rfc7525 here). The last time I looked, there were something like 50 RFCs that are referencing that [*The Register* checked this https://datatracker.ietf.org/doc/rfc7525/referencedby/ list, provided by Farrell – it seems to be close to 70 already].

That's modernising things – the people who did some e-mail protocol 10 or 15 years ago, they may have referenced the then-current version of TLS. Old-fashioned ways of doing things.

That's being used in order to provide a common reference: as people update their implementations, they'll reference a more modern version of TLS, currently TLS 1.2, and as TLS 1.3 is finished, we have an automated-ish way of getting those updates percolating through to the documentation sets.

That's quite successful, I think, because it normalises and updates and modernises a bunch of recommendations.

As that gets into implementations, I hope … it will mean that we have fewer of these surprises about old, laggard versions of TLS being used in bad ways in applications.

DPRIVE: is a different case. DPRIVE is at the point where the RFC for how to do DNS over TLS is done, there are implementations available, and that's at the point where people can experiment with how can they use this in their networks, and does it have any good impact or bad impact on their operations?

The Register: For example, http://www.theregister.co.uk/2016/11/22/dns_boffins_offer_up_privacy_test/ Stubby – announced at IETF 97.

Stephen Farrell: The gist is that DNS privacy is something that is ready to experiment with. The current work in DPRIVE was how to [secure] the hop between and the next DNS provider you talk to.

That's an easy problem to tackle – you talk to that DNS resolver a lot, and you have some shared space, so the overhead of doing the crypto stuff is nowhere.

What happens there, is if you want to resolve theregister.co.uk – your computer will talk to over the DNS protocol to the next resolver upstream (it could be the one at your ISP, or it could be one of the Google public ones, whatever you choose).

DPRIVE will protect that link, because you use it often, and we can amortise the cost of the crypto.

Assuming that [the ISP] needs to find "where is

theregister.co.uk?", he'll eventually talk to the UK ccTLD, and then he'll go talk to .co.uk and then he'll go talk to theregister.co.uk – it's forking the communications a lot more, and it's a little harder to see how to efficiently amortise the crypto.

The DRPIVE working group are now examining whether they think they can produce some technology that will work for that part of the problem.

TCPINC: they're close to finishing, I hope. I think we're close to having some TCP-crypt-based RFCs issued, there's been code for that all along. Whether or not we'll get much deployment of that, we'll see.

We have some use-cases for it – I think there are a bunch of applications that maybe wouldn't be visible to the general public.

Let's say you have an application server that has to run over a socket – an application that runs on top of the Linux kernel, say, where you have to use the kernel because of the interfaces involved, and you can't provide the security above the kernel because you need it inside.

That's where TCPINC fits in. Storage – they have really complex interfaces between the network-available storage server and the kernel, and there's lots of complex distributed processing going on.

In the likes of NetApp and EMC and so on.

For some of those folks, being able to slot in security inside the

kernel, with TCPINC, is attractive. Some, I might expect, will adopt that sort of thing – but it may never be seen on the public Internet.

But my first answer to your question – TLS 1.3, HTTP2 and QUIC are indirect results of RFC 7258, influenced by that, but they're more important.

The Register: Documentation – in a project of this size, touching so much of the documentation of the Internet, the documentation must be hell. We have to keep all the documents moving, updating, and synchronised.

Stephen Farrell: Yeah … mmm … yeah. The IETF produces about 400 pages every two weeks … I have a counter I keep running, averaged over the last six years.

We don't have a goal of forcing overall overall consistency in the RFC series, all the time.

One way of thinking about this: one of the reasons the Web works well is 404!

A certain amount of inconsistency can be what gets you to work, as opposed to not working.

Typically what happens is we would have a bunch of specifications that are sitting around for a number of years, implementations that are happily working according to those specifications – and somebody says “it's time to update that”.

When there's interest, and when there's the likelihood that will change the implementations – as opposed to just being

somebody's good idea – that's when we'll start work on it, and that's when we want to make sure the new documentation is aligned with things like RFC 7258 and so on. And other things that have happened in the meantime.

We re-started a working group on PGP, about two years ago, because people felt they wanted to update the use of crypto in things like PGP.

Part of that involves talking about the wonderful key IDs that people use in PGP, and considering whether any changes to those should be made because of RFC 7258 – because it's an identifier that's stable over time, it could be used for tracking, and so on.

So they've had that discussion, and I think in that case the answer was "it would be more counter-productive to change than the benefit you'd get from changing to something that was harder to track as a key ID".

We don't go back over all 8,000 RFCs and say "who's going to fix this one today?" – it would be hard to motivate people. I did actually try to do some of that, but that didn't work.

The Register: In the business of getting people to move from a new specification to implementation – we know how hard it is to make that change. IPv6 is an example – what are the triggers that make people decide to make the change (get rid of the old SSLs and old TLSs) – where do you get the impetus to go from document to code?

Stephen Farrell: The motivations vary. One great example is to do with HTTP2 and QUIC: there's a whole bunch of people

motivated to use TLS almost ubiquitously, not only because they care about privacy, but because of performance: it moves the point of control back towards the endpoint, not the middle of the network.

One of the interesting and fun things of trying to improve the security properties and privacy properties of the network is that it changes who controls what.

If you encrypt a session, nobody in the middle can do something like inject advertising.

That makes some people happy, and it upsets some other people.

That's a real motivator by the ones made happy by that kind of change, because it reasserts the end-to-end argument in a pretty strong way. If you do the crypto right, then the middlebox can't jump in and modify things – at least not without being detectable.

That's a big motivation.

Another – engineers hate producing crap. People don't like putting out code that's got horrible vulnerabilities.

Not every developer is so noble – but a bunch of them are. People don't like finding out that they're giving imperfect technology to others to use – particularly in people producing open source toolkits like OpenSSL.

They really want to do a good job … moving on as it becomes available to TLS 1.3 because it has better security properties,

and so on.

The Register: I don't think [Tim Hudson and Eric Young] in the late 1990s believed they'd be building something that would last forever.

Stephen Farrell: Yes … there's an interesting study to be done, what are the influences that cause those kinds of effects, that something like SSL becomes as ubiquitous. And how do you start a change? We see things like BoringSSL and other implementations, how they change over time.

There's that motivation that engineers like stuff that works well, which I wouldn't underestimate.

But there's also a whole bunch of corporate network operators not wanting people to misbehave in our networks. If we can improve security and privacy, we can put down the kinds of misbehaviour for those who operate networks, for those who distribute products, or have to do recalls, or would like to avoid future legislation that might be unwise. And so on.

The Register: Putting control back at the edge. As a Unix greybeard would say, you cannot have a secure network, you can only have secure endpoints. A lot of what you're saying is adopting a basic security principle of having secure endpoints.

Stephen Farrell: "You can only have" is overstated. I tend to not be absolutist in these things.

There are some hard problems: inbound malware detection, particularly in an enterprise network where you have people bringing their own devices.

That's a tough one for a sysadmin, if we push everything absolutely to the endpoints.

But it's also true that we can push a lot to the endpoints.

That doesn't mean we can't detect oddities in the network, just by looking at the ciphertext that passes by.

You don't have to be doing deep packet inspection to say “this network is behaving slightly oddly”. Distributed denial of service attacks, for example, those are easy to spot without looking at the plaintext.

Similarly – I'm sitting at my home network right now. I would love a way to understand the traffic that my kids devices were sending out.

I think there's an interesting set of research questions there, as to how I could usefully do that, that makes sense for the average person at home. It should be doable – but that's a research topic.

I agree that pushing intelligence to the edges of the network seems wiser, our experience shows that seems to be the case. Equally, commerce also has shown many times people seeing a business opportunity doing functionality inside the networks.

The real downside of having middleboxes doing things is that they kind of freeze what you're doing, and prevent you innovating.

One of the reasons people did HTTP2 implementations, that

only ever talk ciphertext, is because they found a lot of middleboxes would break the connection if they saw anything that wasn't HTTP 1.1.

In other words, the cleartext had the effect that the middleboxes, that were frozen in time, would prevent the edges from innovating. Once they encrypted the HTTP2 traffic, the middleboxes were willing to say "it's TLS so I won't go near it", and the innovation can kick off again at the edges.

The Register: If we say "going forward, what are the hard problems that are currently moving more slowly than we'd like?"

Stephen Farrell: The big one is on Slide 22 – people do collaborate in various ways with the corporate collection of data.

I suspect that it's very hard to see that changing when we still have an advertising-driven business model for most of the Internet.

I personally think the advertising-driven business model will change – there's no reason why a particular business model has to last forever.

As long as we have that advertising business model, it's hard to see how we make a dramatic improvement in privacy. We can make some improvements, but how we make it dramatically better – it's hard. The incentives are aligned to make all the service providers want to be privacy-unfriendly, from the point of "me", but not perhaps the point of view of 99 per cent of people who use the Internet, and seem happy enough with it.

Data leaks and breaches, database leaks and so on – will mitigate that to some extent, in that they'll cause service providers to realise that storing everything forever is toxic in the end, and they'll get caught by it.

As long as they're advertising-driven, they are motivated to try and find out as much as possible, and tell it to other people.

The Register: If we look at crypto work happening now, we will still see contributions arising from the NSA. That got a lot of criticism a few years ago, when RSA got burned by that kind of contribution. Have processes changed, to make sure that where people are contributing to crypto and user-side security, their contribution is in the interests of users?

Stephen Farrell: There's not an obvious, easy answer.

From the IETF perspective – we are open, you don't need a funny handshake.

As an open organisation, we need to be open to technical contributions from anywhere, be that an employee of the NSA, or be that – as we've had in one case – a teenager from the Ukraine who was commenting on RFCs five or six years ago.

That's a good thing.

The way to think about this is twofold: one is that people are aware of this, and clearly that's sharpened by things like DUAL-EC.

We still see things like Juniper showing up with this kind of

problem, just this year, and we'll probably see more of those. Not all, necessarily, thanks to one organisation – maybe thanks to more than one organisation!

My way of thinking about this is that it's not something we can really capture very well with processes – not with formal processes, that say "you must have n people".

Socially, you can handle it much better.

What we've done, for example: we have the CFRG, the Crypto Forum Research Group, in the Internet Research Task Force. We haven't beefed it up, because it's all volunteer effort – but we've put a bunch of effort into improving the interactions between academic cryptographers and the standards community, through CFRG and some very good work from some of the chairs – Alexey Melnikov and Kenny Paterson.

For example, we organised a workshop on TLS 1.3 where we had academic cryptographers looking at the protocol, because that's important.

In February, the CFRG had just put in place a panel of cryptographers to provide advice when someone turns up and says "here's my algorithm spec". Do we know from the public literature if this is good looking? Been around for a while? Been reviewed? Is it novel and therefore maybe not ready for prime time?

So in a lightweight-process kind of way, we're trying to do that.

The other way of considering that particular problem is that it's very similar to the problems we have with patents.

A bad actor can abuse patent systems and open standards processes in all sorts of fun way. They can have a patent, they can pay a consultant to do their talking for them and not tell the consultant (officially) they have a patent. You can do tricks.

All those strategies would be available to a bad actor who was going to get the community to use some undesirable cryptographic algorithm or construct.

In the IETF context – we're open, we don't have membership, we don't have a way of having someone sign up to a particular patent licensing thing, so we end up with a kind of weird policy on patents.

I think the effect of worrying about somebody trying to convince us to take in borked crypto is kind of the same. We have to be able to think about it in the same way.

There's no process that would stop a determined attacker from at least attempting to get bad stuff into specifications, and they don't have to wear an improper badge to do it. They can hire consultants, etcetera, etcetera.

The number of NSA employees that attend IETF metrics – I don't think it's a useful metric at all. I think how well peoples' contributions are examined is a much more useful metric, and there, things like having the CFRG, having academic cryptographers interacting much more with the standards community – those are more effective ways of doing that.

We've set up a thing called the Advanced Networking Research Prize, which is a prize for already-published

academic work. It pays for the academic come to an IETF meeting, give us a talk, get them involved.

One of the co-chairs of the CFRG, Kenny Parterson, first came as part of an ANRP prize, and now is helping out and chairing the CFRG.

And we now have a workshop for that.

One of the things the IETF did not do as well as we should have 15 years ago – we reacted to the AES competition too optimistically.

NIST did a fantastic job with the AES competition – they did a super-good job of it. We just relied on them too much afterwards, as it happened.

We needed to, and since – in the last two or three years – have rebuilt and built connections between people working on the standards in the IETF, with the academic community, which is also much much larger than it was 20 years ago.

It's easier to do, because there's a lot more academics we can rely on.

One of the things I said about this problem: we in the open Internet community need to not fall into the same fallacy as the intelligence community have fallen into.

We should not think people are guilty by association. That's a fallacy – if you believe that NSA employees are not allowed to contribute, you're making the same mistake they're making.