Some Recent Developments in Internet Security and Privacy

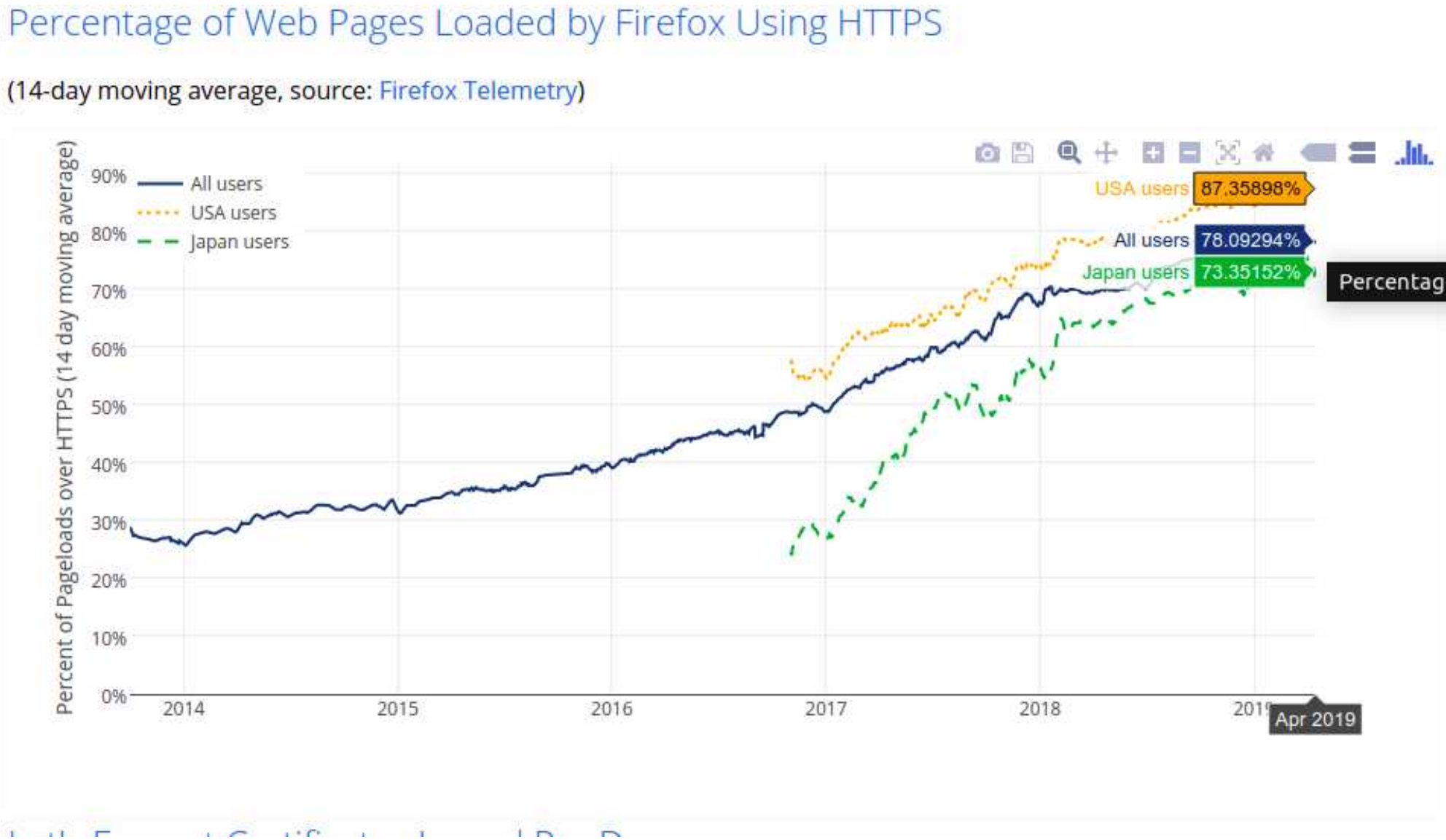
Stephen Farrell
stephen.farrell@cs.tcd.ie
May 2019
(minor updates Nov 2019)

Background/Overview

- Me: Trinity College Dublin, School of Computer Science and Statistics
  - Research topics: security, privacy, delay-tolerant networking
  - Member of Internet Architecture Board (https://iab.org/)
- Overview: Since Snowden popped up, we're doing better with comsec, but need more progress with privacy, centralisaion and legal-but-adversarial applications
- These slides are also at: https://down.dsg.cs.tcd.ie/may19/

Some Better News

https://letsencrypt.org/stats/



Some Middling News

- The Internet community are deprecating “old” versions of TLS (1.0 and 1.1)
  - https://tools.ietf.org/html/draft-ietf-tls-oldversions-deprecate
- Mozilla run a test of ~1M domains to see who’s still stuck on those (i.e. doesn’t support TLS1.2 or TLS1.3)
  - https://hacks.mozilla.org/2019/05/tls-1-0-and-1-1-removal-update/
- There were 12 “.ie” domains in the May 2019 “carnage” list ;-(
  - anpost.ie cso.ie dublinbus.ie museum.ie ulsterbank.ie teanglann.ie bordbia.ie medicalcouncil.ie checktheregister.ie tradesmen.ie pac.ie meathchronicle.ie
- Nov 2019 update: 6 left – anpost.ie, dublinbus.ie, ulsterbank.ie, museum.ie, medicalcouncil.ie, meathchronicle.ie
- Full list at: http://tlscanary-plot-8e95d89854d73f4d.elb.us-west-2.amazonaws.com/tlsdeprecation-carnage.txt
  (which very oddly doesn’t seem to be available via TLS;-)
- Take away: you can and should measure your Internet posture, and even if you don’t, other people will (both good and bad actors)

More Middling News
TLS Sessions per test site

Do53	site	N	min	max	stdev	avg
	ietf.org	22	4	12	2.12	9.29
	irishtimes.com	22	74	**158**	22.47	126.36
	jell.ie	22	4	12	1.96	6.73
	nytimes.com	22	29	**98**	16.07	81.23
	rte.ie	19	38	63	6.33	50.32
	tcd.ie	22	69	**102**	10.61	92.18
	www.ietf.org	14	4	7	0.73	5.07

DoT	site	N	min	max	stdev	avg
	ietf.org	34	15	34	6.14	25.03
	irishtimes.com	37	129	**196**	15.03	157.11
	jell.ie	38	12	30	5.69	21.11
	nytimes.com	34	76	**133**	12.44	105.44
	rte.ie	27	47	98	13.02	67.48
	tcd.ie	34	98	**123**	5.33	112.53

- Do53 = DNS over 53 (clear)
- DoT = stubby with ~10 recursive IPs configured

- N = count of PCAP files
- Min,max,stdev,avg refer to the number of TLS sessions in each PCAP

- Selenium mostly used FF, but some chrome/opera
- Browsers/selenium drivers are “out of the box” with no special config, nor plug-ins, extensions etc.

Some Bad News

- There's too much from which to chose;-(
- I recently wrote a sadly dystopian document arguing that we need an extended threat model for the Internet
  - https://tools.ietf.org/html/draft-farrell-etm
  - Updated with co-author: https://tools.ietf.org/html/draft-arkko-farrell-arch-model-t-00
- Basic argument: current threat model (RFC3552) assumes protocol endpoints are not compromised, so we develop comsec tools (like TLS) to mitigate threats from middleboxes and attackers
- BUT, there are many examples of endpoints that are adversarial and we all encounter those all the time
  - E.g. commercial surveillance, phishing, malware from "app stores"
- Not clear what'll happen with this, but some folks are starting to work on the topic

Internet Engineering Task Force

- IETF (https://ietf.org/) is the body that defines Internet protocols like IP, TCP, HTTP, SMTP, TLS etc
- IETF documents protocols in the RFC (request for comment) series, which has been running since 1969; 8600+ RFCs have been issued over that 50 year period
- IETF has no members – all you need to participate successfully is an email address and a sound technical opinion
  - Well, you also need a *lot* of patience;-)
  - No members => no membership fee, and no voting, instead we operate via “rough consensus”
- Pretty much all work in the IETF is done openly on mailing lists or at (virtual or f2f) meetings to which anyone can come
  - Remote participation at f2f meetings is even getting pretty good
- Exceptions to openness mainly relate to personnel matters, e.g. the nominations committee (10 randomly selected active participants) select the leadership

IETF and Security/Privacy

- All RFCs contain a “security considerations” section and (today) undergo a fairly good security review before being finalised
  - See RFC3552
- Privacy issues are more and more a part of that review
  - See RFC6973
- Post-Snowden, the IETF has done a lot of work to try improve the bits of the Internet that it can help improve…

Pervasive Monitoring

From RFC7258/BCP188: “Pervasive Monitoring is an Attack”

> Pervasive Monitoring (PM) is widespread (and often covert) surveillance through intrusive gathering of protocol artefacts, including application content, or protocol meta-data such as headers. Active or passive wiretaps and traffic analysis, (e.g., correlation, timing or measuring packet sizes), or subverting the cryptographic keys used to secure protocols can also be used as part of pervasive monitoring. PM is distinguished by being indiscriminate and very large-scale, rather than by introducing new types of technical compromise.

PM is not everything

- PM is far from the only security or privacy issue on which we need to work
  - Spam, malware, DDoS, …
  - But mitigations for PM can also help a lot with other problems
- Hypothesis: If we work to address PM, and prioritise services and mechanisms that mitigate PM and that are also effective against other attacks then we will be doing the “right thing”
- And the “we” there means us all, not just bodies like the IETF

2014 IAB Statement

"We recommend that encryption be deployed throughout the protocol stack since there is not a single place within the stack where all kinds of communication can be protected.

The IAB urges protocol designers to design for confidential operation by default. We strongly encourage developers to include encryption in their implementations, and to make them encrypted by default. We similarly encourage network and service operators to deploy encryption where it is not yet deployed, and we urge firewall policy administrators to permit encrypted traffic."

https://www.iab.org/2014/11/14/iab-statement-on-internet-confidentiality/

2019 IAB Statement

“Avoiding Unintended Harm to Internet Infrastructure”

“Many legislative and regulatory efforts around the world are currently focusing on how to preserve the Internet’s benefits to society while curtailing undesirable uses.

While members of the technical community that designs, builds, and operates the Internet have a variety of positions regarding these efforts, the Internet Architecture Board (IAB) has a responsibility to inform legislators and regulators when proposals might have impacts on the viability of the Internet as a whole, considering all of its uses and users.

This statement discusses possible unintended effects policy and regulatory proposals may have on the Internet. It develops our recent submission regarding Australia’s Assistance and Access Bill 2018 into a more general overview of the potential impacts of such proposals, along with our recommendations for avoiding adverse outcomes. …”

- https://www.iab.org/2019/09/04/iab-statement-on-avoiding-unintended-harm-to-internet-infrastructure/

Security & Privacy vs. “Management”

- There’s a tension (ack’d in RFC7258) that the better we protect security/privacy, the less well (some) network management tools work
  - Classic example: DPI for firewalls
  - There are also elements of mis-trust between some stakeholders that make this harder
- This recurs over and over, and generates angst:
  - Via efforts to break TLS (“Pretty please standardise my MitM technique”)
  - Via efforts to encrypt protocol data units that are relevant for network management causing heartache for network management folks
- Hard to know to what extent this is due to lazy/legacy n/w management techniques and how much it’ll be a lasting problem, but the increasing prevalence of ciphertext will not change IMO
  - In other words: we need to develop new n/w management tools that still work well when networks carry much more ciphertext

Security, Privacy and Law Enforcement

- Adding better security and privacy features to the network also changes the game somewhat for law enforcement
- Some of what used be accessible becomes harder to access or even “impossible” to access (in practice)
- It’s very hard to see how to square the circle between law enforcement requirements and the technical requirements for better security and privacy that are both equally real
  - Esp. when in a largely open-source world
- I’d love to see a discussion about those kinds of requirements (not mechanisms, requirements) but it’s hard to see that happening
  - I wrote a paper about that a while back: "Requirements Analysis Required--Otherwise Targeted Monitoring Enables Pervasive Monitoring." Computer 49.3 (2016): 34-40.
    - Extended form (without paywall:-) https://down.dsg.cs.tcd.ie/cpus/

Examples of this Tussle

- We'll look at two topical examples:
  - DNS over HTTPS
  - Encrypted SNI
- In both cases we'll see that efforts to improve comsec cause issues for current operators
- We don't yet know how either of these will turn out...

DNS Privacy (1)

- DNS is the Domain Name System
  - Maps names like tcd.ie to IP addresses (stored in A/AAAA records in the DNS)
  - Also used for other things, e.g. anti-spam blocklists, DMARC/DKIM/SPF
- DNSSEC extended DNS to provide data origin authentication, but isn't widely deployed (about 1% of zones are signed)
- DNS, even with DNSSEC, provides no confidentiality since DNS data is, by design, public
- BUT, the act of querying a DNS domain is sensitive (see RFC7626):
  - if I lookup "somebaddisease.org" I might prefer that my ISP or employer doesn't get to know about that
  - If someone looks at the set of DNS queries emitted by a device it may be possible to (re-)identify the user at the device, even if all other traffic is encrypted/ignored
  - The set of DNS queries I (or my home) emits over time is none of your business and ought not be recorded for-profit
- Full answer: probably use Tor Browser, but we also now have DNS privacy mechanisms that allow me to do DNS over TLS

DNS Privacy (2)

- There's more than one variant for how to do this
- DNS over TLS (DoT) – run DNS over TCP using TLS on port 853 (RFC7858)
  - Cleartext/traditional DNS is sometimes now called Do53 for "DNS on port 53"
- DNS over HTTPS (DoH) – run DNS over HTTPS on port 443 (RFC8484)
- Why two?
  - DoT is a good way to incrementally improve DNS service – just add a port 853 listener to your existing DNS infrastructure
  - DoH is a good way to improve DNS privacy and performance for browsers in a way that's less easy to censor as it mixes the DNS traffic with other HTTPS traffic

DNS Centralisation

- Traditional DNS model was for stubs to get their recursive (server) IP address via configuration or DHCP, so you end up using your network provider's choice of DNS server
- There are now "public" DNS servers (recursives) operated by large companies such as Google (quad8) or Cloudflare (quad1) or smaller specialists like quad9
  - These tend to pick "cute" IP addresses like 8.8.8.8 etc. and that has been used as a counter-censorship tool
  - These can also do filtering, e.g. of "bad" DNS names, some do, some don't
- The success of these DNS recursive resolver services has lead to (more) fear of centralisation
  - Apparently about 14% of DNS queries use Google's service

DNS Centralisation + DoH

- Mozilla have announced a policy for supporting DoH in Firefox
  - https://wiki.mozilla.org/Security/DOH-resolver-policy
- Basically, they plan to configure some DoH services that meet their policy and try use those for DNS name resolution
  - Various conditions on trying DoH vs. Do53 exist in their code
  - First and currently only service instance is Cloudflare (sortof == quad1)
- Many operators have (loudly) voiced many concerns
  - Moves “choice” from n/w operator to application; breaks split-DNS; works-around DNS censorship/filtering (netnanny); …
  - Even the UK house of lords getting in on the act: https://hansard.parliament.uk/Lords/2019-05-14/debates/E84CBBAE-E005-46E0-B7E5-845882DB1ED8/InternetEncryption
  - And the UK house of commons too: https://www.parliament.uk/business/publications/written-questions-answers-statements/written-questions-answers/?page=1&max=20&questiontype=AllQuestions&house=commons&member=1463
- My take: every time we see a widening of crypto deployment someone yells that the sky is falling. The sky hasn’t fallen. But there are issues to be figured out, esp split-DNS and what “choice” even means here when appoximately zero percent of users even know the DNS exists
- Nov 2019: Chrome and Microsoft have announced their own variants of Mozilla’s DoH plans.

TLS and Encrypted SNI

- Transport Layer Security (TLS, RFC8446) is the security protocol that secures the web and many other applications – HTTP running over TLS is what makes HTTPS
- One web server instance (e.g. an apache install using VirtualHost) can, and very frequently does, serve multiple web sites
- Each of those may (and is v. likely to) use different TLS server key pairs/certificates
- When using HTTPS, one of the first things the server needs to do is pick which TLS server key/certificate to use for the TLS session
  - As the client needs to verify that it's talking to the correct server via the TLS server certificate, which (for the web) contains the domain name of the web site
- Result: the first TLS message the client sends (the ClientHello) needs to specify the web site (DNS name) for which the TLS session is being established
  - That's done using the Server Name Indication (SNI) extension to the ClientHello

SNI as a leak

- Since the SNI value is sent in the first message, nobody has a key with which to encrypt anything yet, so SNI is sent in cleartext
- When accessing https://bank.example.com/getBalance the “getBalance” part will be encrypted (later, when the full HTTP request is sent) but the DNS name (“bank.example.com”) is sent in clear in the SNI extension
- That’s a noticeable leak, especially as the SNI is visible to everyone on the path (my ISP, the site’s ISP, every intermediate router, hosters, governments) and SNI has also been used for censorship
  - https://www.bleepingcomputer.com/news/security/south-korea-is-censoring-the-internet-by-snooping-on-sni-traffic/
- So-called domain “fronting” has been used in the past (where the SNI has the hoster’s name but the HTTP request has the “real” DNS name) but, from an engineering standpoint, that’s very brittle and (we surmise) some servers were forced to turn that off in some jurisdictions
- Previously, we weren’t motivated to try fix this quite tricky problem, as the DNS name was also sent in clear via Do53, but now, with DoT/DoH, it’s useful to do better, so we’re working on that

Encrypted SNI (ESNI)

- The IETF's TLS working group reached consensus on the problem to be addressed
  - https://tools.ietf.org/html/draft-ietf-tls-sni-encryption
- And a solution to the problem is being developed:
  - https://tools.ietf.org/html/draft-ietf-tls-esni
  - Early days, likely 12-18 months from being done but there is a Firefox nightly build with support for the -02 version draft and Cloudflare have deployed the server side for the same thing, and there's plenty of interest, so this is reasonably likely to get fairly wide deployment
- Full disclosure: I'm working on an implementation of that for the OpenSSL library and have a contract from OTF to fund that work, see https://defo.ie/
  - Whether or not any of my code ends up in a standard release of the OpenSSL library is for later and will be determined via the OpenSSL project's usual mechanisms for handling contributions

How does ESNI work?

- Basic idea is that the web site publishes a new public key in the DNS (“ESNIKeys”) and an ESNI aware client (e.g. browser) can check if that exists and if it does, then use it to send the SNI value in encrypted form in a new TLS ClientHello extension
  - The fact that ESNI is being used may still be visible, but we’re also considering countering that leak via “greasing” - which means having browsers that are not using ESNI sending a ClientHello that looks like it does use ESNI
- Another value (a “cover” or “public” name) may be sent in clear in the existing TLS ClientHello SNI extension
  - Sort of an officially supported version of domain fronting
- So I could setup a TLS session for HTTP that appears to be with “example.com” but is actually with “bank.example.com”
  - You can imagine that that excites people who like the idea and who dislike the idea!
  - But there’s more:-)

ESNI and multiple services (1)

- Web sites use different content delivery networks (CDNs) like cloudflare and akamai and some web sites switch between those (for price or availability reasons) very frequently
- CDNs are highly likely to provide the most common deployment of ESNI, as they want to be in the business of offering “cover” for their customers and they also allow web sites to “hide in the crowd” which is a centralising tendency that many privacy mechanisms share
- To make the DNS scale as it needs to, DNS information is cached - all values returned from the DNS have a time-to-live (TTL) value in seconds, which can be small (e.g. 5 seconds) or large (e.g. 1 week)
- When playing the ESNI game, a client needs to access both the IP address of the web site (the A/AAAA DNS records) and the new public key (the ESNIKeys resource record)
- Those could get out of whack for various reasons, perhaps especially if a web site is switching between CDNs
  - BUT – we don’t yet know how bad this problem may be – very early evidence seems to show it’ll not be a big deal for clients, but could occasionally be for web sites

ESNI and multiple services (2)

- Two proposals have been made to address this problem:
  - Include IP address information in the ESNIKeys value
  - Include IP address-range information in the ESNIKeys value
- The former is the proposal in the current draft specification
- If that ends up being the final approach, then it'll cause some sysadmins to freak (again:-) as it'd mean duplication of address information in different places in the DNS, which'd likely break some tooling and network monitoring
  - And it might only work for some kinds of address management in some CDNs
  - And it's an architectural no-no too IMO;-)
- The address-range proposal seems better to me, but is more complex and risks leaving some cases where the A/AAAA and ESNIKeys values might not match
  - Effect could be failed connections or falling back to cleartext SNI when ESNI should be usable
- Nov 2019: DNS and TLS handshake elements of this undergoing change (ESNIKeys → HTTPSSVC and ESNI→ ECHO) but getting there...
- Current state is that implementers are just starting to play with this so it's not yet set off alarm bells
  - But it will;-) Probably in early 2020.

Conclusion

- We've seen good progress in communications security since 2013 and that's continuing
- We're seeing some tussles between various players as the consequences of that improved comsec gets deployed and changes the network from "mostly plaintext" towards "mostly ciphertext"
- Those tussles will continue
- We're starting to see people more seriously considering how protocols affect centralisation and privacy but there are some hard questions there for which we don't currently have any real answer

Thanks

Offline questions welcome too
stephen.farrell@cs.tcd.ie

These slides:
https://down.dsg.cs.tcd.ie/may19/
Updated at https://github.com/sftcd/witidtm