Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 7, 2020

S. Farrell
Trinity College Dublin

July 6, 2019

We're gonna need a bigger threat model
draft-farrell-etm-03

Abstract

We argue that an expanded threat model is needed for Internet protocol development as protocol endpoints can no longer be considered to be generally trustworthy for any general definition of "trustworthy."

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 7, 2020.

Table of Contents

1. Introduction

[[There's a github repo for this -- issues and PRs are welcome there. <https://github.com/sftcd/etm>]]

[RFC3552], Section 3 defines an "Internet Threat Model" which has been commonly used when developing Internet protocols. That assumes that "the end-systems engaging in a protocol exchange have not themselves been compromised." RFC 3552 is a formal part of of the IETF's process as it is also BCP72.

Since RFC 3552 was written, we have seen a greater emphasis on considering privacy and [RFC6973] provides privacy guidance for protocol developers. RFC 6973 is not a formal BCP, but appears to have been useful for protocol developers as it is referenced by 38 later RFCs at the time of writing [1].

BCP188, [RFC7258] subsequently recognised pervasive monitoring as a particular kind of attack and has also been relatively widely referenced (39 RFCs at the time of writing [2]). To date, perhaps most documents referencing BCP188 have considered state-level or in-network adversaries.

In this document, we argue that we need to epxand our threat model to acknowledge that today many applications are themselves rightly considered potential adversaries for at least some relevant actors. However, those (good) actors cannot in general refuse to communicate and will with non-negligible probability encounter applications that are adversarial.

We also argue that not recognising this reality causes Internet protocol designs to sometimes fail to protect the systems and users who depend on those.

Discussion related to expanding our concept of threat-model ought not (but perhaps inevitably will) involve discussion of weakening how confidentiality is provided in Internet protocols. Whilst it may superficially seem to be the case that encouraging in-network interception could help with detection of adversarial application behaviours, such a position is clearly mistaken once one notes that adding middleboxes that can themselves be adversarial cannot be a solution to the problem of possibly encountering adversarial code on the network. It is also the case that the IETF has rough consensus to provide better, and not weaker, security and privacy, which includes confidentiality services. The IETF has maintained that consensus over three decades, despite repeated (and repetitive;-) debates on the topic. That consensus is represented in [RFC2804], BCP 200 [RFC1984] and more latterly, the above-mentioned BCP 188 as

well as in the numerous RFCs referencing those works. The probability that discussion of expanding our threat model leads to a change in that rough consensus seems highly remote.

However, it is not clear if the IETF will reach rough consensus on a description of such an expanded threat model. We argue that ignoring this aspect of deployed reality may not bode well for Internet protocol development.

Absent such an expanded threat model, we expect to see more of a mismatch between expectaions and the deployment reality for some Internet protocols.

Version -02 of this internet-draft was a submission to the IAB's DEDR workshop [3]. We note that another author independently proposed changes to the Internet threat model for related, but different, reasons, [I-D.arkko-arch-internet-threat-model] also as a submission to the DEDR workshop.

We are saddened by, and apologise for, the somewhat dystopian impression that this document may impart - hopefully, there's a bit of hope at the end;-)

2. Examples of deliberate adversarial behaviour in applications

In this section we describe a few documented examples of deliberate adversarial behaviour by applications that could affect Internet protocol development. The adversarial behaviours described below involve various kinds of attack, varying from simple fraud, to credential theft, surveillance and contributing to DDoS attacks. This is not intended to be a comprehensive nor complete survey, but to motivate us to consider deliberate adversarial behaviour by applications.

While we have these examples of deliberate adversarial behaviour, there are also many examples of application developers doing their best to protect the security and privacy of their users or customers. That's just the same as the case today where we need to consider in-network actors as potential adversaries despite the many examples of network operators who do act primarily in the best interests of their users. So this section is not intended as a slur on all or some application developers.

2.1. Malware in curated application stores

Despite the best efforts of curators, so-called App-Stores frequently distribute malware of many kinds and one recent study [curated] claims that simple obfuscation enables malware to avoid detection by

even sophisticated operators. Given the scale of these deployments, ditribution of even a small percentage of malware-infected applictions can affect a huge number of people.

2.2. Virtual private networks (VPNs)

Virtual private networks (VPNs) are supposed to hide user traffic to various degrees depending on the particular technology chosen by the VPN provider. However, not all VPNs do what they say, some for example misrepresenting the countries in which they provide vantage points. [vpns]

2.3. Compromised (home) networks

What we normally might consider network devices such as home routers do also run applications that can end up being adversarial, for example running DNS and DHCP attacks from home routers targeting other devices in the home. One study [home] reports on a 2011 attack that affected 4.5 million DSL modems in Brazil. The absence of software update [RFC8240] has been a major cause of these issues and rises to the level that considering this as intentional behaviour by device vendors who have chosen this path is warranted.

2.4. Web browsers

Tracking of users in order to support advertising based business models is ubiquitous on the Internet today. HTTP header fields (such as cookies) are commonly used for such tracking, as are structures within the content of HTTP responses such as links to 1x1 pixel images and (ab)use of Javascript APIs offered by browsers. [tracking]

While some people may be sanguine about this kind of tracking, others consider this behaviour unwelcome, when or if they are informed that it happens, [attitude] though the evidence here seems somewhat harder to interpret and many studies (that we have found to date) involve small numbers of users. Historically, browsers have not made this kind of tracking visible and have enabled it by default, though some recent browser versions are starting to enable visibility and blocking of some kinds of tracking. Browsers are also increasingly imposing more stringent requirements on plug-ins for varied security reasons.

2.5. Web site policy deception

Many web sites today provide some form of privacy policy and terms of service, that are known to be mostly unread. [unread] This implies that, legal fiction aside, users of those sites have not in reality agreed to the specific terms published and so users are therefore

highly exposed to being exploited by web sites, for example [cambridge] is a recent well-publicised case where a service provider abused the data of 87 million users via a partnership. While many web site operators claim that they care deeply about privacy, it seems prudent to assume that some (or most?) do not in fact care about user privacy, or at least not in ways with which many of their users would agree. And of course, today's web sites are actually mostly fairly complex web applications and are no longer static sets of HTML files, so calling these "web sites" is perhaps a misnomer, but considered as web applications, that may for example link in advertising networks, it seems clear that many exist that are adversarial.

2.6. Tracking bugs in mail

Some mail user agents (MUAs) render HTML content by default (with a subset not allowing that to be turned off, perhaps particularly on mobile devices) and thus enable the same kind of adversarial tracking seen on the web. Attempts at such intentional tracking are also seen many times per day by email users - in one study [mailbug] the authors estimated that 62% of leakage to third parties was intentional, for example if leaked data included a hash of the recipient email address.

2.7. Troll farms in online social networks

Online social network applications/platforms are well-known to be vulnerable to troll farms, sometimes with tragic consequences, [4] where organised/paid sets of users deliberately abuse the application platform for reasons invisible to a normal user. For-profit companies building online social networks are well aware that subsets of their "normal" users are anything but. In one US study, [troll] sets of troll accounts were roughly equally distributed on both sides of a controversial discussion. While Internet protocol designers do sometimes consider sybil attacks [sybil], arguably we have not provided mechanisms to handle such attacks sufficiently well, especially when they occur within walled-gardens. Equally, one can make the case that some online social networks, at some points in their evolution, appear to have prioritised counts of active users so highly that they have failed to invest sufficient effort for detection of such troll farms.

2.8. Smart televisions

There have been examples of so-called "smart" televisions spying on their owners without permission [5] and one survey of user attitudes [smarttv] found "broad agreement was that it is unacceptable for the data to be repurposed or shared" although the level of user

understanding may be questionable. What is clear though is that such devices generally have not provided controls for their owners that would allow them to meaningfully make a decision as to whether or not they want to share such data.

2.9. So-called Internet of things

Many so-called Internet of Things (IoT) devices ("so-called" as all devices were already things:-) have been found extremely deficient when their security and privacy aspects were analysed, for example children's toys. [toys] While in some cases this may be due to incompetence rather than being deliberately adversarial behaviour, the levels of incompetence frequently seen imply that it is valid to consider such cases as not being accidental.

2.10. Attacks leveraging compromised high-level DNS infrastructure

Recent attacks [6] against DNS infrastructure enable subsequent targetted attacks on specific application layer sources or destinations. The general method appears to be to attack DNS infrastructure, in these cases infrastructure that is towards the top of the DNS naming hierarchy and "far" from the presumed targets, in order to be able to fake DNS responses to a PKI, thereby acquiring TLS server certificates so as to subsequently attack TLS connections from clients to services (with clients directed to an attacker-owned server via additional fake DNS responses).

Attackers in these cases seem well resourced and patient - with "practice" runs over months and with attack durations being infrequent and short (e.g. 1 hour) before the attacker withdraws.

These are sophisticated multi-protocol attacks, where weaknesses related to deployment of one protocol (DNS) bootstrap attacks on another protocol (e.g. IMAP/TLS), via abuse of a 3rd protocol (ACME), partly in order to capture user IMAP login credentials, so as to be able to harvest message store content from a real message store.

The fact that many mail clients regularly poll their message store means that a 1-hour attack is quite likely to harvest many cleartext passwords or crackable password hashes. The real IMAP server in such a case just sees fewer connections during the "live" attack, and some additional connections later. Even heavy email users in such cases that might notice a slight gap in email arrivals, would likely attribute that to some network or service outage.

In many of these cases the paucity of DNSSEC-signed zones (about 1% of existing zones) and the fact that many resolvers do not enforce

DNSSEC validation (e.g., in some mobile operating systems) assisted the attackers.

It is also notable that some of the personnel dealing with these attacks against infrastructure entites are authors of RFCs and Internet-drafts. That we haven't provided protocol tools that better protect against these kinds of attack ought hit "close to home" for the IETF.

In terms of the overall argument being made here, the PKI and DNS interactions, and the last step in the "live" attack all involve interaction with a deliberately adversarial application. Later, use of acquired login credentials to harvest message store content involves an adversarial client application. It all cases, a TLS implementation's PKI and TLS protocol code will see the fake endpoints as protocol-valid, even if, in the real world, they are clearly fake. This appears to be a good argument that our current threat model is lacking in some respect(s), even as applied to our currently most important security protocol (TLS).

2.11. BGP hijacking

There is a clear history of BGP hijacking [bgphijack] being used to ensure endpoints connect to adversarial applications. As in the previous example, such hijacks can be used to trick a PKI into issuing a certificate for a fake entity. Indeed one study [hijackdet] used the emergence of new web server TLS key pairs during the event, (detected via Internet-wide scans), as a distinguisher between one form of deliberate BGP hijacking and indadvertent route leaks.

3. Inadvertent adversarial behaviours

Not all adversarial behaviour by applications is deliberate, some is likely due to various levels of carelessness (some quite understandable, others not) and/or due to erroneous assumptions about the environments in which those applications (now) run.
We very briefly list some such cases:

- Application abuse for command and control, for example, use of IRC or apache logs for malware command and control [7]

- Carelessly leaky buckets [8], for example, lots of Amazon S3 leaks showing that careless admins can too easily cause application server data to become available to adversaries

- Virtualisation exposing secrets, for example, Meltdown and Spectre [9] and similar side-channels

- Compromised badly-maintained web sites, that for example, have led to massive online databases of passwords [10]

- Supply-chain attacks, for example, the Target attack [11] or malware within pre-installed applications on Android phones. [bloatware]

- Breaches of major service providers, that many of us might have assumed would be sufficiently capable to be the best large-scale "Identity providers", for example:

  * 3 billion accounts: yahoo [12]

  * "up to 600M" account passwords stored in clear: facebook [13]

  * many millions at risk: telcos selling location data [14]

  * 50 million accounts: facebook [15]

  * 14 million accounts: verizon [16]

  * "hundreds of thousands" of accounts: google [17]

  * unknown numbers, some email content exposed: microsoft [18]

- Breaches of smaller service providers: Too many to enumerate, sadly

4. Possible directions for an expanded threat model

As we believe useful conclusions in this space require community consensus, we won't offer definitive descriptions of an expanded threat model but we will call out some potential directions that could be explored as one follow-up to the DEDR workshop and thereafter, if there is interest in this topic.

Before doing so, it is worth calling out one of the justifications for the RFC 3553 definition of the Internet threat model which is that going beyond an assumption that protocol endpoints have not been compromised rapidly introduces complexity into the analysis. We do have plenty of experience that when security and privacy solutions add too much complexity and/or are seen to add risks without benefits, those tend not to be deployed. One of the risks in expanding our threat model that we need to recognise is that the end result could be too complex, might not be applied during protocol design, or worse, could lead to flawed risk analyses. One of the constraints on work on an expanded threat model is therefore that the

result has to remain usable by protocol designers who are not security or privacy experts.

4.1. Develop a BCP for privacy considerations

It may be time for the IETF to develop a BCP for privacy considerations, possibly starting from [RFC6973].

4.2. Consider the user perspective

[I-D.nottingham-for-the-users] argues that, in relevant cases where there are conflicting requirements, the "IETF considers end users as its highest priority concern." Doing so seems consistent with the expanded threat model being argued for here, so may indicate that a BCP in that space could also be useful.

4.3. Consider ABuse-cases as well as use-cases

Protocol developers and those implementing and deploying Internet technologies are typically most interested in a few specific use-cases for which they need solutions. Expanding our threat model to include adversarial application behaviours [abusecases] seems likely to call for significant attention to be paid to potential abuses of whatever new or re-purposed technology is being considered.

4.4. Re-consider protocol design "lore"

It could be that this discussion demonstrates that it is timely to reconsider some protocol design "lore" as for example is done in [I-D.iab-protocol-maintenance]. More specifically, protocol extensibility mechanisms may inadvertently create vectors for abuse-cases, given that designers cannot fully analyse their impact at the time a new protocol is defined or standardised. One might conclude that a lack of extensibility could be a virtue for some new protocols, in contrast to earlier assumptions. As pointed out by one commenter though, people can find ways to extend things regardless, if they feel the need.

4.5. Isolation

Sophisticated users can sometimes deal with adversarial behaviours in applications by using different instances of those applications, for example, differently configured web browsers for use in different contexts. Applications (including web browsers) and operating systems are also building in isolation via use of different processes or sandboxing. Protocol artefacts that relate to uses of such isolation mechanisms might be worth considering. To an extent, the IETF has in practice already recognised some of these issues as being

in-scope, e.g. when considering the linkability issues with mechanisms such as TLS session tickets, or QUIC connection identifiers.

4.6. Transparency

Certificate transparency (CT) [RFC6962] has been an effective countermeasure for X.509 certificate mis-issuance, which used be a known application layer misbehaviour in the public web PKI. CT can also help with post-facto detection of some infrastructure attacks where BGP or DNS weakenesses have been leveraged so that some certification authority is tricked into issuing a certificate for the wrong entity.

While the context in which CT operates is very constrained (essentially to the public CAs trusted by web browsers), similar approaches could perhaps be useful for other protocols or technologies.

In addition, legislative requirements such as those imposed by the GDPR for subject access to data [19] could lead to a desire to handle internal data structures and databases in ways that are reminiscent of CT, though clearly with significant authorisation being required and without the append-only nature of a CT log.

4.7. Minimise

As recommended in [RFC6973] data minimisation and additional encryption are likely to be helpful - if applications don't ever see data, or a cleartext form of data, then they should have a harder time misbehaving. Similarly, not adding new long-term identifiers, and not exposing existing ones, would seem helpful.

4.8. Same-Origin Policy

The Same-Origin Policy (SOP) [RFC6454] perhaps already provides an example of how going beyond the RFC 3552 threat model can be useful. Arguably, the existence of the SOP demonstrates that at least web browsers already consider the 3552 model as being too limited. (Clearly, differentiating between same and not-same origins implicitly assumes that some origins are not as trustworthy as others.)

4.9. Greasing

The TLS protocol [RFC8446] now supports the use of GREASE [I-D.ietf-tls-grease] as a way to mitigate on-path ossification. While this technique is not likely to prevent any deliberate

misbehaviours, it may provide a proof-of-concept that network protocol mechanisms can have impact in this space, if we spend the time to try analyse the incentives of the various parties.

4.10. Generalise OAuth Threat Model

The OAuth threat model [RFC6819] provides an extensive list of threats and security considerations for those implementing and deploying OAuth version 2.0 [RFC6749]. That document is perhaps too detailed to serve as useful generic guidance but does go beyond the Internet threat model from RFC3552, for example it says:

> two of the three parties involved in the OAuth protocol may collude to mount an attack against the 3rd party. For example, the client and authorization server may be under control of an attacker and collude to trick a user to gain access to resources.

It could be useful to attempt to derive a more abstract threat model from that RFC that considers threats in more generic multi-party contexts.

4.11. One (or more) endpoint may be compromised

The quote from OAuth above also has another aspect - it considers the effect of compromised endpoints on those that are not compromised. It may therefore be interesting to consider the consequeneces that would follow from this OLD/NEW change to RFC3552

OLD: In general, we assume that the end-systems engaging in a protocol exchange have not themselves been compromised.

NEW:

In general, we assume that one of the protocol engines engaging in a protocol exchange has not been compromised at the run-time of the exchange.

4.12. Look again at how well we're securing infrastructure

Some attacks (e.g. against DNS or routing infrastructure) appear to benefit from current infrastructure mechanisms not being deployed, e.g. DNSSEC, RPKI. In the case of DNSSEC, deployment is still minimal despite much time having elapsed. This suggests a number of different possible avenues for investigation:

- For any protocol dependent on infrastructure like DNS or BGP, we ought analysse potential outcomes in the event the relevant infrastructure has been compromised

- Protocol designers perhaps ought consider post-facto detection compromise mechanisms in the event that it is infeasible to mitigate attacks on infrastructure that is not under local control

- Despite the sunk costs, it may be worth re-considering infrastructure security mechanisms that have not been deployed, and hence are ineffective.

4.13. Consider recovery from attack as part of protocol design

Recent work on multiparty messaging security primitives [I-D.ietf-mls-architecture] considers "post-compromise security" as an inherent part of the design of that protocol. Perhaps protocol designers ought generally consider recovery from attack during protocol design - we do know that all widely used protocols will at sometime be subject to successful attack, whether that is due to deployment or implementation error, or, as is less common, due to protocol design flaws.

4.14. Don't think in terms of hosts

More and more, protocol endpoints are not being executed on what used be understood as a host system. The web and Javascript model clearly differs from traditional host models, but so do most server-side deployments these days, thanks to virtualisation.

As yes unpublished work on this topic within the IAB stackevo [20] programme, appears to posit the same kind of thesis. In the stackevo case, that work would presumably lead to some new definition of protocol endpoint, but (consensus on) such a definition may not be needed for an expanded threat model. For this work, it may be sufficient to note that protocol endpoints can no longer be considered to be executing on a traditional host, to assume (at protocol design time) that all endpoints will be run in a virtualised environment where co-tenants and (sometimes) hypervisors are adversaries, and to then call for analysis of such scenarios.

5. Conclusions

At this stage we don't think it approriate to claim that any strong conclusion can be reached based on the above. We do however, claim that the is a topic that could be worth discussion as part of the follow-up to at the DEDR workshop and more generally within the IETF.

6. Security Considerations

This draft is all about security, and privacy.

Encryption is one of the most effective tools in countering network based attackers and will also have a role in protecting against adversarial applications. However, today many existing tools for countering adversarial applications assume they can inspect network traffic to or from potentially adversarial applications. These facts of course cause tensions (e.g. see [RFC8404]). Expanding our threat model could possibly help reduce some of those tensions, if it leads to the development of protocols that make exploitation harder or more transparent for adversarial applications.

7. IANA Considerations

There are no IANA considerations.

8. Acknowledgements

With no implication that they agree with some or all of the above, thanks to Jari Arkko, Carsten Bormann, Christian Huitema and Daniel Kahn Gillmor for comments on an earlier version of the text.

Thanks to Jari Arkko, Ted Hardie and Brian Trammell for discussions on this topic after they (but not the author) had attended the DEDR workshop.

9. References

9.1. Informative References

[abusecases]
McDermott, J. and C. Fox, "Using abuse case models for security requirements analysis", IEEE Annual Computer Security Applications Conference (ACSAC'99) 1999, 1999, <https://www.acsac.org/1999/papers/wed-b-1030-john.pdf>.

[attitude]
Chanchary, F. and S. Chiasson, "User Perceptions of Sharing, Advertising, and Tracking", Symposium on Usable Privacy and Security (SOUPS) 2015, 2015, <https://www.usenix.org/conference/soups2015/proceedings/presentation/chanchary>.

[bgphijack]
Sermpezis, P., Kotronis, V., Dainotti, A., and X. Dimitropoulos, "A survey among network operators on BGP prefix hijacking", ACM SIGCOMM Computer Communication Review 48, no. 1 (2018): 64-69., 2018, <https://arxiv.org/pdf/1801.02918.pdf>.

[bloatware]
Gamba, G., Rashed, M., Razaghpanah, A., Tapiado, J., and N. Vallina-Rodriguez, "An Analysis of Pre-installed Android Software", arXiv preprint arXiv:1905.02713 (2019)., 2019, <https://arxiv.org/pdf/1905.02713.pdf>.

[cambridge]
Isaak, J. and M. Hanna, "User Data Privacy: Facebook, Cambridge Analytica, and Privacy Protection", Computer 51.8 (2018): 56-59, 2018, <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8436400>.

[curated] Hammad, M., Garcia, J., and S. MaleK, "A large-scale empirical study on the effects of code obfuscations on Android apps and anti-malware products", ACM International Conference on Software Engineering 2018, 2018, <https://www.ics.uci.edu/~seal/publications/2018ICSE_Hammad.pdf>.

[hijackdet]
Schlamp, J., Holz, R., Gasser, O., Korste, A., Jacquemart, Q., Carle, G., and E. Biersack, "Investigating the nature of routing anomalies: Closing in on subprefix hijacking attacks", International Workshop on Traffic Monitoring and Analysis, pp. 173-187. Springer, Cham, 2015., 2015, <https://www.net.in.tum.de/fileadmin/bibtex/publications/papers/schlamp_TMA_1_2015.pdf>.

[I-D.arkko-arch-internet-threat-model]
Arkko, J., "Changes in the Internet Threat Model", draft-arkko-arch-internet-threat-model-00 (work in progress), April 2019.

[I-D.iab-protocol-maintenance]
Thomson, M., "The Harmful Consequences of the Robustness Principle", draft-iab-protocol-maintenance-03 (work in progress), May 2019.

[I-D.ietf-mls-architecture]
Omara, E., Beurdouche, B., Rescorla, E., Inguva, S., Kwon, A., and A. Duric, "The Messaging Layer Security (MLS) Architecture", draft-ietf-mls-architecture-02 (work in progress), March 2019.

[I-D.ietf-tls-grease]
Benjamin, D., "Applying GREASE to TLS Extensibility", draft-ietf-tls-grease-02 (work in progress), January 2019.

[I-D.nottingham-for-the-users]
Nottingham, M., "The Internet is for End Users", draft-nottingham-for-the-users-08 (work in progress), June 2019.

[mailbug] Englehardt, S., Han, J., and A. Narayanan, "I never signed up for this! Privacy implications of email tracking", Proceedings on Privacy Enhancing Technologies 2018.1 (2018): 109-126., 2018, <https://www.degruyter.com/downloadpdf/j/popets.2018.2018.issue-1/popets-2018-0006/popets-2018-0006.pdf>.

[RFC1984] IAB and IESG, "IAB and IESG Statement on Cryptographic Technology and the Internet", BCP 200, RFC 1984, DOI 10.17487/RFC1984, August 1996, <https://www.rfc-editor.org/info/rfc1984>.

[RFC2804] IAB and IESG, "IETF Policy on Wiretapping", RFC 2804, DOI 10.17487/RFC2804, May 2000, <https://www.rfc-editor.org/info/rfc2804>.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <https://www.rfc-editor.org/info/rfc3552>.

[RFC6454] Barth, A., "The Web Origin Concept", RFC 6454, DOI 10.17487/RFC6454, December 2011, <https://www.rfc-editor.org/info/rfc6454>.

[RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, DOI 10.17487/RFC6749, October 2012, <https://www.rfc-editor.org/info/rfc6749>.

[RFC6819] Lodderstedt, T., Ed., McGloin, M., and P. Hunt, "OAuth 2.0 Threat Model and Security Considerations", RFC 6819, DOI 10.17487/RFC6819, January 2013, <https://www.rfc-editor.org/info/rfc6819>.

[RFC6962] Laurie, B., Langley, A., and E. Kasper, "Certificate Transparency", RFC 6962, DOI 10.17487/RFC6962, June 2013, <https://www.rfc-editor.org/info/rfc6962>.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <https://www.rfc-editor.org/info/rfc6973>.

[RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <https://www.rfc-editor.org/info/rfc7258>.

[RFC8240] Tschofenig, H. and S. Farrell, "Report from the Internet of Things Software Update (IoTSU) Workshop 2016", RFC 8240, DOI 10.17487/RFC8240, September 2017, <https://www.rfc-editor.org/info/rfc8240>.

[RFC8404] Moriarty, K., Ed. and A. Morton, Ed., "Effects of Pervasive Encryption on Operators", RFC 8404, DOI 10.17487/RFC8404, July 2018, <https://www.rfc-editor.org/info/rfc8404>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[smarttv] Malkin, N., Bernd, J., Johnson, M., and S. Egelman, ""What Can't Data Be Used For?" Privacy Expectations about Smart TVs in the U.S.", European Workshop on Usable Security (Euro USEC) 2018, 2018, <https://www.ndss-symposium.org/wp-content/uploads/2018/06/eurousec2018_16_Malkin_paper.pdf>.

[sybil] Viswanath, B., Post, A., Gummadi, K., and A. Mislove, "An analysis of social network-based sybil defenses", ACM SIGCOMM Computer Communication Review 41(4), 363-374. 2011, 2011, <https://conferences.sigcomm.org/sigcomm/2010/papers/sigcomm/p363.pdf>.

[toys] Chu, G., Apthorpe, N., and N. Feamster, "Security and Privacy Analyses of Internet of Things Childrens' Toys", IEEE Internet of Things Journal 6.1 (2019): 978-985., 2019, <https://arxiv.org/pdf/1805.02751.pdf>.

[tracking] Ermakova, T., Fabian, B., Bender, B., and K. Klimek, "Web Tracking-A Literature Review on the State of Research", Proceedings of the 51st Hawaii International Conference on System Sciences, 2018, <https://scholarspace.manoa.hawaii.edu/bitstream/10125/50485/paper0598.pdf>.

[troll] Stewart, L., Arif, A., and K. Starbird, "Examining trolls and polarization with a retweet network", ACM Workshop on Misinformation and Misbehavior Mining on the Web 2018, 2018, <https://faculty.washington.edu/kstarbi/examining-trolls-polarization.pdf>.

[unread] Obar, J. and A. Oeldorf-Hirsch, "The biggest lie on the internet: Ignoring the privacy policies and terms of service policies of social networking services", Information, Communication and Society (2018): 1-20, 2018, <https://doi.org/10.1080/1369118X.2018.1486870>.

[vpns] Khan, M., DeBlasio, J., Voelker, G., Snoeren, A., Kanich, C., and N. Vallina-Rodrigue, "An empirical analysis of the commercial VPN ecosystem", ACM Internet Measurement Conference 2018 (pp. 443-456), 2018, <https://eprints.networks.imdea.org/1886/1/imc18-final198.pdf>.

9.2. URIs

[1] https://datatracker.ietf.org/doc/rfc6973/referencedby/

[2] https://datatracker.ietf.org/doc/rfc7258/referencedby/

[3] https://www.iab.org/activities/workshops/dedr-workshop/

[4] https://www.nytimes.com/2018/10/20/us/politics/saudi-image-campaign-twitter.html

[5] https://www.welivesecurity.com/2013/11/22/lg-admits-that-its-smart-tvs-have-been-watching-users-and-transmitting-data-without-consent/

[6] https://krebsonsecurity.com/2019/02/a-deep-dive-on-the-recent-widespread-dns-hijacking-attacks/

[7] https://security.stackexchange.com/questions/100577/creating-botnet-cc-server-what-architecture-should-i-use-irc-http

[8] https://businessinsights.bitdefender.com/worst-amazon-breaches

[9] https://www.us-cert.gov/ncas/alerts/TA18-004A

[10] https://haveibeenpwned.com/Passwords

[11] https://www.zdnet.com/article/how-hackers-stole-millions-of-credit-card-records-from-target/

[12] https://www.wired.com/story/yahoo-breach-three-billion-accounts/

[13] https://www.pcmag.com/news/367319/facebook-stored-up-to-600m-user-passwords-in-plain-text

[14] https://www.zdnet.com/article/us-telcos-caught-selling-your-location-data-again-senator-demands-new-laws/

[15] https://www.cnet.com/news/facebook-breach-affected-50-million-people/

[16] https://www.zdnet.com/article/millions-verizon-customer-records-israeli-data/

[17] https://www.wsj.com/articles/google-exposed-user-data-feared-repercussions-of-disclosing-to-public-1539017194

[18] https://motherboard.vice.com/en_us/article/ywyz3x/hackers-could-read-your-hotmail-msn-outlook-microsoft-customer-support

[19] https://gdpr-info.eu/art-15-gdpr/

[20] https://github.com/stackevo/endpoint-draft/blob/master/draft-trammell-whats-an-endpoint.md

Appendix A. Change Log

This isn't gonna end up as an RFC, but may as well be tidy...

A.1. Changes from -02 to -03

- Integrated some changes based on discussion with Ted, Jari and Brian.

A.2. Changes from -01 to -02

- Oops - got an RFC number wrong in reference

A.3. Changes from -00 to -01

- Made a bunch more edits and added more references
- I had lots of typos (as always:-)
- cabo: PR#1 fixed more typos and noted extensbility danger

Author's Address

Stephen Farrell
Trinity College Dublin

Email: stephen.farrell@cs.tcd.ie