Network Working Group
Internet-Draft
Intended status: Informational
Expires: June 17, 2020

J. Arkko
Ericsson
S. Farrell
Trinity College Dublin

December 15, 2019

Challenges and Changes in the Internet Threat Model
draft-arkko-farrell-arch-model-t-01

Abstract

Communications security has been at the center of many security improvements in the Internet. The goal has been to ensure that communications are protected against outside observers and attackers.

This memo suggests that the existing threat model, while important and still valid, is no longer alone sufficient to cater for the pressing security issues in the Internet. For instance, it is also necessary to protect systems against endpoints that are compromised, malicious, or whose interests simply do not align with the interests of the users. While such protection is difficult, there are some measures that can be taken.

It is particularly important to ensure that as we continue to develop Internet technology, non-communications security related threats, and privacy issues, are properly understood.

Copyright Notice

Table of Contents

1. Introduction

Communications security has been at the center of many security improvements in the Internet. The goal has been to ensure that communications are protected against outside observers and attackers. At the IETF, this approach has been formalized in BCP 72 [RFC3552], which defined the Internet threat model in 2003.

The purpose of a threat model is to outline what threats exist in order to assist the protocol designer. But RFC 3552 also ruled some threats to be in scope and of primary interest, and some threats out of scope [RFC3552]:

> The Internet environment has a fairly well understood threat model. In general, we assume that the end-systems engaging in a protocol exchange have not themselves been compromised. Protecting against an attack when one of the end-systems has been compromised is extraordinarily difficult. It is, however, possible to design protocols which minimize the extent of the damage done under these circumstances.
>
> By contrast, we assume that the attacker has nearly complete control of the communications channel over which the end-systems communicate. This means that the attacker can read any PDU (Protocol Data Unit) on the network and undetectably remove, change, or inject forged packets onto the wire.

However, the communications-security -only threat model is becoming outdated. This is due to three factors:

o Advances in protecting most of our communications with strong cryptographic means. This has resulted in much improved communications security, but also highlights the need for addressing other, remaining issues. This is not to say that communications security is not important, it still is: improvements are still needed. Not all communications have been protected, and even out of the already protected communications, not all of their aspects have been fully protected. Fortunately, there are ongoing projects working on improvements.

o Adversaries have increased their pressure against other avenues of attack, from compromising devices to legal coercion of centralized endpoints in conversations.

o New adversaries and risks have arisen, e.g., due to creation of large centralized information sources.

o While communications-security does seem to be required to protect privacy, more is needed.

In short, attacks are migrating towards the currently easier targets, which no longer necessarily include direct attacks on traffic flows. In addition, trading information about users and ability to influence them has become a common practice for many Internet services, often without users understanding those practices.

This memo suggests that the existing threat model, while important and still valid, is no longer alone sufficient to cater for the pressing security issues in the Internet. For instance, while it continues to be very important to protect Internet communications against outsiders, it is also necessary to protect systems against endpoints that are compromised, malicious, or whose interests simply do not align with the interests of the users.

Of course, there are many trade-offs in the Internet on who one chooses to interact with and why or how. It is not the role of this memo to dictate those choices. But it is important that we understand the implications of different practices. It is also important that when it comes to basic Internet infrastructure, our chosen technologies lead to minimal exposure with respect to the non-communications threats.

It is particularly important to ensure that non-communications security related threats are properly understood for any new Internet technology. While the consideration of these issues is relatively new in the IETF, this memo provides some initial ideas about potential broader threat models to consider when designing protocols for the Internet or when trying to defend against pervasive monitoring. Further down the road, updated threat models could result in changes in BCP 72 [RFC3552] (guidelines for writing security considerations) and BCP 188 [RFC7258] (pervasive monitoring), to include proper consideration of non-communications security threats.

It may also be necessary to have dedicated guidance on how systems design and architecture affect security. The sole consideration of communications security aspects in designing Internet protocols may lead to accidental or increased impact of security issues elsewhere.

For instance, allowing a participant to unnecessarily collect or receive information may lead to a similar effect as described in [RFC8546] for protocols: over time, unnecessary information will get used with all the associated downsides, regardless of what deployment expectations there were during protocol design.

This memo does not stand alone. To begin with, it is a merge of earlier work by the two authors [I-D.farrell-etm] [I-D.arkko-arch-internet-threat-model]. There are also other documents discussing this overall space, e.g. [I-D.lazanski-smart-users-internet] [I-D.arkko-arch-dedr-report].

The authors of this memo envisage independent development of each of those (and other work) with an eventual goal to extract an updated (but usefully brief!) description of an extended threat model from the collection of works. We consider it an open question whether this memo, or any of the others, would be usefully published as an RFC.

The rest of this memo is organized as follows. Section 2 makes some observations about the situation, with respect to communications security and beyond. The section also provides a number of real-world examples.

Section 3 discusses some high-level implications that can be drawn, such as the need to consider what the "ends" really are in an "end-to-end" communication.

Section 4 discusses some potential remedies, both from the point of view of a system design, as well as from the point of IETF procedures and recommended analysis procedures when designing new protocols. For instance, Section 4.1 will also discuss high-level guidance to addressing these threats, and Section 4.3.4 suggests some potential changes to the definition of the IETF's "Internet Threat Model". It is also apparent that the dangers posed by pervasive monitoring need to be taken in a broader light, given the evolution of the threats beyond communications security.

Comments are solicited on these and other aspects of this document. The best place for discussion is on the arch-discuss list (https://www.ietf.org/mailman/listinfo/Architecture-discuss).

Finally, Section 5 draws some conclusions for next steps.

2. Observations

2.1. Communications Security Improvements

Being able to ask about threat model improvements is due to progress already made: The fraction of Internet traffic that is cryptographically protected has grown tremendously in the last few years. Several factors have contributed to this change, from Snowden revelations to business reasons and to better available technology such as HTTP/2 [RFC7540], TLS 1.3 [RFC8446], QUIC [I-D.ietf-quic-transport].

In many networks, the majority of traffic has flipped from being cleartext to being encrypted. Reaching the level of (almost) all traffic being encrypted is no longer something unthinkable but rather a likely outcome in a few years.

At the same time, technology developments and policy choices have driven the scope of cryptographic protection from protecting only the pure payload to protecting much of the rest as well, including far more header and meta-data information than was protected before. For instance, efforts are ongoing in the IETF to assist encrypting transport headers [I-D.ietf-quic-transport], server domain name information in TLS [I-D.ietf-tls-esni], and domain name queries [RFC8484].

There have also been improvements to ensure that the security protocols that are in use actually have suitable credentials and that those credentials have not been compromised, see, for instance, Let's Encrypt [RFC8555], HSTS [RFC6797], HPKP [RFC7469], and Expect-CT [I-D.ietf-httpbis-expect-ct].

This is not to say that all problems in communications security have been resolved - far from it. But the situation is definitely different from what it was a few years ago. Remaining issues will be and are worked on; the fight between defense and attack will also continue. Communications security will stay at the top of the agenda in any Internet technology development.

2.2. Beyond Communications Security

There are, however, significant issues beyond communications security in the Internet. To begin with, it is not necessarily clear that one can trust all the endpoints in any protocol interaction.

Of course, the endpoints were never trusted, but the pressures against endpoints issues seem to be mounting. For instance, the users may not be in as much control over their own devices as they

used to be due to manufacturer-controlled operating system installations and locked device ecosystems. And within those ecosystems, even the applications that are available tend to have privileges that users by themselves would most likely not desire those applications have, such as excessive rights to media, location, and peripherals. There are also designated efforts by various authorities to hack end-user devices as a means of intercepting data about the user.

The situation is different, but not necessarily better on the side of servers. The pattern of communications in today's Internet is almost always via a third party that has at least as much information as the other parties have. For instance, these third parties are typically endpoints for any transport layer security connections, and able to see any communications or other messaging in cleartext. There are some exceptions, of course, e.g., messaging applications with end-to-end protection.

With the growth of trading users' information by many of these third parties, it becomes necessary to take precautions against endpoints that are compromised, malicious, or whose interests simply do not align with the interests of the users.

Specifically, the following issues need attention:

o Security of users' devices and the ability of the user to control their own equipment.

o Leaks and attacks related to data at rest.

o Coercion of some endpoints to reveal information to authorities or surveillance organizations, sometimes even in an extra-territorial fashion.

o Application design patterns that result in cleartext information passing through a third party or the application owner.

o Involvement of entities that have no direct need for involvement for the sake of providing the service that the user is after.

o Network and application architectures that result in a lot of information collected in a (logically) central location.

o Leverage and control points outside the hands of the users or end-user device owners.

For instance, while e-mail transport security [RFC7817] has become much more widely distributed in recent years, progress in securing

e-mail messages between users has been much slower. This has lead to a situation where e-mail content is considered a critical resource by some mail service providers who use the content for machine learning, advertisement targeting, and other purposes unrelated to message delivery. Equally however, it is unclear how some useful anti-spam techniques could be deployed in an end-to-end encrypted mail universe (with today's end-to-end mail sercurity protocols).

The Domain Name System (DNS) shows signs of ageing but due to the legacy of deployed systems has changed very slowly. Newer technology [RFC8484] developed at the IETF enables DNS queries to be performed confidentially, but its initial deployment is happening mostly in browsers that use global DNS resolver services, such as Cloudflare's 1.1.1.1 or Google's 8.8.8.8. This results in faster evolution and better security for end users.

However, if one steps back and considers the potential security and privacy effects of these developments, the outcome could appear different. While the security and confidentiality of the protocol exchanges improves with the introduction of this new technology, at the same time this could lead to a move from using a worldwide distributed set of DNS resolvers into a far smaller set of centralised global resolvers. While these resolvers are very well maintained (and a great service), they are potential high-value targets for pervasive monitoring and Denial-of-Service (DoS) attacks. In 2016, for example, DoS attacks were launched against Dyn, one of the largest DNS providers, leading to some outages. It is difficult to imagine that DNS resolvers wouldn't be a target in many future attacks or pervasive monitoring projects.

Unfortunately, there is little that even large service providers can do to not be a DDoS target, not to refuse authority-sanctioned pervasive monitoring. As a result it seems that a reasonable defense strategy may be to aim for outcomes where such highly centralised control points are unecessary or don't handle sensitive data. (Recalling that with the DNS, information about the requestor and the act of requesting an answer are what is potentially sensitive, rather than the content of the answer.)

There are other examples of the perils of centralised solutions in Internet infrastructure. The DNS example involves an interesting combination of information flows (who is asking for what domain names) as well as a potential ability to exert control (what domains will actually resolve to an address). Routing systems are primarily about control. While there are intra-domain centralized routing solutions (such as PCE [RFC4655]), a control within a single administrative domain is usually not the kind of centralization that we would be worried about. Global centralization would be much more

concerning. Fortunately, global Internet routing is performed a among peers. However, controls could be introduced even in this global, distributed system. To secure some of the control exchanges, the Resource Public Key Infrastructure (RPKI) system ([RFC6480]) allows selected Certification Authorities (CAs) to help drive decisions about which participants in the routing infrastructure can make what claims. If this system were globally centralized, it would be a concern, but again, fortunately, current designs involve at least regional distribution.

In general, many recent attacks relate more to information than communications. For instance, personal information leaks typically happen via information stored on a compromised server rather than capturing communications. There is little hope that such attacks can be prevented entirely. Again, the best course of action seems to be avoid the disclosure of information in the first place, or at least to not perform that in a manner that makes it possible that others can readily use the information.

2.3. Examples

2.3.1. 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.

2.3.1.1. 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, distribution of even a small percentage of malware-infected applictions can affect a huge number of people.

2.3.1.2. 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].

2.3.1.3. 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.

2.3.1.4. 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.

2.3.1.5. 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.

2.3.1.6. 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.

2.3.1.7. Troll farms in online social networks

Online social network applications/platforms are well-known to be vulnerable to troll farms, sometimes with tragic consequences 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.

2.3.1.8. Smart televisions

There have been examples of so-called "smart" televisions spying on their owners 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.

2.3.1.9. Internet of things

Internet of Things (IoT) devices (which might be "so-called Internet of Things" as all devices were already things:-) have been found 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 these aspects have simply not been considered a priority.

2.3.1.10. Attacks leveraging compromised high-level DNS infrastructure

Recent attacks [DeepDive] 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).

2.3.1.11. 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.

2.3.2. 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:

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

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

- o Virtualisation exposing secrets, for example, Meltdown and Spectre [MeltdownAndSpectre] [Kocher2019] [Lipp2018] and other similar side-channel attacks.

- o Compromised badly-maintained web sites, that for example, have led to massive online [Passwords].

- o Supply-chain attacks, for example, the [TargetAttack] or malware within pre-installed applications on Android phones [Bloatware].

- o 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: https://www.wired.com/story/yahoo-breach-three-billion-accounts/

  * "up to 600M" account passwords stored in clear: https://www.pcmag.com/news/367319/facebook-stored-up-to-600m-user-passwords-in-plain-text

  * many millions at risk: https://www.zdnet.com/article/us-telcos-caught-selling-your-location-data-again-senator-demands-new-laws/

  * 50 million accounts: https://www.cnet.com/news/facebook-breach-affected-50-million-people/

  * 14 million accounts: https://www.zdnet.com/article/millions-verizon-customer-records-israeli-data/

  * "hundreds of thousands" of accounts: https://www.wsj.com/articles/google-exposed-user-data-feared-repercussions-of-disclosing-to-public-1539017194

  * unknown numbers, some email content exposed: https://motherboard.vice.com/en_us/article/ywyz3x/hackers-could-read-your-hotmail-msn-outlook-microsoft-customer-support

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

3. Analysis

3.1. The Role of End-to-end

[RFC1958] notes that "end-to-end functions can best be realised by end-to-end protocols":

> The basic argument is that, as a first principle, certain required end-to-end functions can only be performed correctly by the end-systems themselves. A specific case is that any network, however carefully designed, will be subject to failures of transmission at some statistically determined rate. The best way to cope with this is to accept it, and give responsibility for the integrity of

communication to the end systems. Another specific case is end-to-end security.

The "end-to-end argument" was originally described by Saltzer et al [Saltzer]. They said:

> The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the endpoints of the communication system. Therefore, providing that questioned function as a feature of the communication system itself is not possible.

These functional arguments align with other, practical arguments about the evolution of the Internet under the end-to-end model. The endpoints evolve quickly, often with simply having one party change the necessary software on both ends. Whereas waiting for network upgrades would involve potentially a large number of parties from application owners to multiple network operators.

The end-to-end model supports permissionless innovation where new innovation can flourish in the Internet without excessive wait for other parties to act.

But the details matter. What is considered an endpoint? What characteristics of Internet are we trying to optimize? This memo makes the argument that, for security purposes, there is a significant distinction between actual endpoints from a user's interaction perspective (e.g., another user) and from a system perspective (e.g., a third party relaying a message).

This memo proposes to focus on the distinction between "real ends" and other endpoints to guide the development of protocols. A conversation between one "real end" to another "real end" has necessarily different security needs than a conversation between, say, one of the "real ends" and a component in a larger system. The end-to-end argument is used primarily for the design of one protocol. The security of the system, however, depends on the entire system and potentially multiple storage, compute, and communication protocol aspects. All have to work properly together to obtain security.

For instance, a transport connection between two components of a system is not an end-to-end connection even if it encompasses all the protocol layers up to the application layer. It is not end-to-end, if the information or control function it carries actually extends beyond those components. For instance, just because an e-mail server can read the contents of an e-mail message does not make it a legitimate recipient of the e-mail.

This memo also proposes to focus on the "need to know" aspect in systems. Information should not be disclosed, stored, or routed in cleartext through parties that do not absolutely need to have that information.

The proposed argument about real ends is as follows:

> Application functions are best realised by the entities directly serving the users, and when more than one entity is involved, by end-to-end protocols. The role and authority of any additional entities necessary to carry out a function should match their part of the function. No information or control roles should be provided to these additional entities unless it is required by the function they provide.

For instance, a particular piece of information may be necessary for the other real endpoint, such as message contents for another user. The same piece of information may not be necessary for any additional parties, unless the information had to do with, say, routing information for the message to reach the other user. When information is only needed by the actual other endpoint, it should be protected and be only relayed to the actual other endpoint. Protocol design should ensure that the additional parties do not have access to the information.

Note that it may well be that the easiest design approach is to send all information to a third party and have majority of actual functionality reside in that third party. But this is a case of a clear tradeoff between ease of change by evolving that third party vs. providing reasonable security against misuse of information.

Note that the above "real ends" argument is not limited to communication systems. Even an application that does not communicate with anyone else than its user may be implemented on top of a distributed system where some information about the user is exposed to untrusted parties.

The implications of the system security also extend beyond information and control aspects. For instance, poorly design component protocols can become DoS vectors which are then used to attack other parts of the system. Availability is an important aspect to consider in the analysis along other aspects.

3.2. Trusted networks

Some systems are thought of as being deployed only in a closed setting, where all the relevant nodes are under direct control of the network administrators. Technologies developed for such networks

tend to be optimized, at least initially, for these environments, and may lack security features necessary for different types of deployments.

It is well known that many such systems evolve over time, grow, and get used and connected in new ways. For instance, the collaboration and mergers between organizations, and new services for customers may change the system or its environment. A system that used to be truly within an administrative domain may suddenly need to cross network boundaries or even run over the Internet. As a result, it is also well known that it is good to ensure that underlying technologies used in such systems can cope with that evolution, for instance, by having the necessary security capabilities to operate in different environments.

In general, the outside vs. inside security model is outdated for most situations, due to the complex and evolving networks and the need to support mixture of devices from different sources (e.g., BYOD networks). Network virtualization also implies that previously clear notions of local area networks and physical proximity may create an entirely different reality from what appears from a simple notion of a local network.

Similarly, even trusted, well-managed parties can be problematic, even when operating openly in the Internet. Systems that collect data from a large number of Internet users, or that are used by a large number of devices have some inherent issues: large data stores attract attempts to use that data in a manner that is not consistent with the users' interests. They can also become single points of failure through network management, software, or business failures. See also [I-D.arkko-arch-infrastructure-centralisation].

3.2.1. Even closed networks can have compromised nodes

This memo argues that the situation is even more dire than what was explained above. It is impossible to ensure that all components in a network are actually trusted. Even in a closed network with carefully managed components there may be compromised components, and this should be factored into the design of the system and protocols used in the system.

For instance, during the Snowden revelations it was reported that internal communication flows of large content providers were compromised in an effort to acquire information from large numbers of end users. This shows the need to protect not just communications targeted to go over the Internet, but in many cases also internal and control communications.

Furthermore, there is a danger of compromised nodes, so communications security alone will be insufficient to protect against this. The defences against this include limiting information within networks to the parties that have a need to know, as well as limiting control capabilities. This is necessary even when all the nodes are under the control of the same network manager; the network manager needs to assume that some nodes and communications will be compromised, and build a system to mitigate or minimise attacks even under that assumption.

Even airgapped networks can have these issues, as evidenced, for instance, by the Stuxnet worm. The Internet is not the only form of connectivity, as most systems include, for instance, USB ports that proved to be the achilles heel of the targets in the Stuxnet case. More commonly, every system runs large amount of software, and it is often not practical or even possible to prevent compromised code even in a high-security setting, let alone in commercial or private networks. Installation media, physical ports, both open source and proprietary programs, firmware, or even innocent-looking components on a circuit board can be suspect. In addition, complex underlying computing platforms, such as modern CPUs with underlying security and management tools are prone to problems.

In general, this means that one cannot entirely trust even a closed system where you picked all the components yourself. Analysis for the security of many interesting real-world systems now commonly needs to include cross-component attacks, e.g., the use of car radios and other externally communicating devices as part of attacks launched against the control components such as breaks in a car [Savage].

3.3. Balancing Threats

Note that not all information needs to be protected, and not all threats can be protected against. But it is important that the main threats are understood and protected against.

Sometimes there are higher-level mechanisms that provide safeguards for failures. For instance, it is very difficult in general to protect against denial-of-service against compromised nodes on a communications path. However, it may be possible to detect that a service has failed.

Another example is from packet-carrying networks. Payload traffic that has been properly protected with encryption does not provide much value to an attacker. For instance, it does not always make sense to encrypt every packet transmission in a packet-carrying system where the traffic is already encrypted at other layers. But

it almost always makes sense to protect control communications and to understand the impacts of compromised nodes, particularly control nodes.

4. Actions

This section discusses potential actions to be taken by the Internet community:

4.1. Basic Guidelines

As [RFC3935] says:

> We embrace technical concepts such as decentralized control, edge-user empowerment and sharing of resources, because those concepts resonate with the core values of the IETF community.

To be more specific, this memo suggests the following guidelines for protocol designers:

1. Consider first principles in protecting information and systems, rather than following a specific pattern such as protecting information in a particular way or at a particular protocol layer. It is necessary to understand what components can be compromised, where interests may or may not be aligned, and what parties have a legitimate role in being a party to a specific information or a control task.

2. Once you have something, do not pass it onto others without serious consideration: In other words, minimize information passed to others. Information passed to another party in a protocol exchange should be minimized to guard against the potential compromise of that party.

3. Perform end-to-end protection via other parties: Information passed via another party who does not intrinsically need the information to perform its function should be protected end-to-end to its intended recipient. This guideline is general, and holds equally for sending TCP/IP packets, TLS connections, or application-layer interactions. As [RFC8546] notes, it is a useful design rule to avoid "accidental invariance" (the deployment of on-path devices that over-time start to make assumptions about protocols). However, it is also a necessary security design rule to avoid "accidental disclosure" where information originally thought to be benign and untapped over-time becomes a significant information leak. This guideline can also be applied for different aspects of security, e.g.,

confidentiality and integrity protection, depending on what the specific need for information is in the other parties.

4. Minimize passing of control functions to others: Any passing of control functions to other parties should be minimized to guard against the potential misuse of those control functions. This applies to both technical (e.g., nodes that assign resources) and process control functions (e.g., the ability to allocate number or develop extensions). Control functions can also become a matter of contest and power struggle, even in cases where their function as such is minimal, as we saw with the IANA transition debates.

5. Avoid centralized resources: While centralized components, resources, and function provide usually a useful function, there are grave issues associated with them. Protocol and network design should balance the benefits of centralized resources or control points against the threats arising from them. The general guideline is to avoid such centralized resources when possible. And if it is not possible, find a way to allow the centralized resources to be selectable, depending on context and user settings.

6. Have explicit agreements: When users and their devices provide information to network entities, it would be beneficial to have an opportunity for the users to state their requirements regarding the use of the information provided in this way. While the actual use of such requirements and the willingness of network entities to agree to them remains to be seen, at the moment even the technical means of doing this are limited. For instance, it would be beneficial to be able to embed usage requirements within popular data formats.

7. Treat parties that your equipment connects to with suspicion, even if the communications are encrypted. The other endpoint may misuse any information or control opportunity in the communication. Similarly, even parties within your own system need to be treated with suspicision, as some nodes may become compromised.

8. Do not take any of this as blanket reason to provide no information to anyone, encrypt everything to everyone, or other extreme measures. However, the designers of a system need to be aware of the different threats facing their system, and deal with the most serious ones (of which there are typically many). Similarly, users should be aware of the choices made in a particular design, and avoid designs or products that protect against some threats but are wide open to other serious issues.

4.2. Potential Further Guidelines

4.2.1. 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.

4.2.2. 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.

4.2.3. 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 [GDPRAccess] 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.

4.2.4. 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.

4.2.5. 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.)

4.2.6. 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.

4.2.7. 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.

4.2.8. 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.

4.2.9. 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.

4.2.10. 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] 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.

4.2.11. Trusted Computing

Various trusted computing mechanisms allow placing some additional trust on a particular endpoint. This can be useful to address some of the issues in this memo:

- A network manager of a set of devices may be assured that the devices have not been compromised.

- An outside party may be assured that someone who runs a device employs a particular software installation in that device, and that the software runs in a protected environment.

IETF work such as TEEP [I-D.ietf-teep-architecture] [I-D.ietf-teep-protocol] and RATS [I-D.ietf-rats-eat] may be helpful in providing attestations to other nodes about a particular endpoint, or lifecycle management of such endpoints.

One should note, however, that it is often not possible to fully protect endpoints (see, e.g., [Kocher2019] [Lipp2018] [I-D.taddei-smart-cless-introduction] [I-D.mcfadden-smart-endpoint-taxonomy-for-cless]). And of course, a trusted computing may be set up and controlled by a party that itself is not trusted; a client that contacts a server that the server's owner runs in a trusted computing setting does not change the fact that the client and the server's owner may have different interests. As a result, there is a need to prepare for the possibility that another party in a communication is not entirely trusted.

4.2.12. Trust Boundaries

Traditional forms of communication equipment have morphed into today's virtualized environments, where new trust boundaries exist, e.g., between different virtualisation layers. And an application might consider itself trusted while not entirely trusting the underlying operating system. A browser application wants to protect itself against Javascript loaded from a website, while the website considers itself and the Javascript an application that it wants to protect from the browser.

In general, there are multiple parties even in a single device, with differing interests, including some that have (or claim to) the interest of the human user in mind.

4.3. Does IETF Analysis of Protocols Need to Change?

It may also be necessary to make procedural changes in how new protocols are defined at the IETF. For instance, our existing documentation of threat models and requirements for security considerations sections may not be adequate in today's world.

These suggested changes are entirely tentative.

4.3.1. Develop a BCP for privacy considerations

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

4.3.2. 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.

4.3.3. 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.

4.3.4. Potential changes in RFC 3552

The earlier quote from OAuth (Section 4.2.7) 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 a change to [RFC3552]. RFC 3552 is the RFC that defines "An Internet Threat Model". It also provides guidance to writing Security Considerations sections in other RFCs. One initial, draft proposal for such changes would be this:

OLD:

In general, we assume that the end-systems engaging in a protocol exchange have not themselves been compromised. Protecting against an attack when one of the end-systems has been compromised is extraordinarily difficult. It is, however, possible to design protocols which minimize the extent of the damage done under these circumstances.

NEW:

In general, we assume that the end-system engaging in a protocol exchange has not itself been compromised. Protecting against an attack of a protocol implementation itself is extraordinarily difficult. It is, however, possible to design protocols which minimize the extent of the damage done when the other parties in a protocol become compromised or do not act in the best interests the end-system implementing a protocol.

In addition, the following new section could be added to discuss the capabilities required to mount an attack:

NEW:

3.x. Other endpoint compromise

In this attack, the other endpoints in the protocol become compromised. As a result, they can, for instance, misuse any information that the end-system implementing a protocol has sent to the compromised endpoint.

System and architecture aspects definitely also need more attention from Internet technology developers and standards organizations. Here is one possible addition:

NEW:

The design of any Internet technology should start from an understanding of the participants in a system, their roles, and the extent to which they should have access to information and ability to control other participants.

4.3.5. Potential Changes in RFC 7258

Other additional guidelines may be necessary also in [RFC7258], the RFC that specifies how IETF work should take into account pervasive monitoring, such as the one performed as a part of broad, indiscriminate surveillance activity.

An initial, draft suggestion for starting point of those changes could be adding the following paragraph after the 2nd paragraph in Section 2:

NEW:

> PM attacks include those cases where information collected by a legitimate protocol participant is misused for PM purposes. The attacks also include those cases where a protocol or network architecture results in centralized data storage or control functions relating to many users, raising the risk of said misuse.

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 and more work.

To start with, Internet technology developers need to be better aware of the issues beyond communications security, and consider them in design. At the IETF it would be beneficial to include some of these considerations in the usual systematic security analysis of technologies under development.

In particular, when the IETF develops infrastructure technology for the Internet (such as routing or naming systems), considering the impacts of data generated by those technologies is important. Minimising data collection from users, minimising the parties who get exposed to user data, and protecting data that is relayed or stored in systems should be a priority.

A key focus area at the IETF has been the security of transport protocols, and how transport layer security can be best used to provide the right security for various applications. However, more work is needed in equivalently broadly deployed tools for minimising or obfuscating information provided by users to other entities, and the use of end-to-end security through entities that are involved in the protocol exchange but who do not need to know everything that is being passed through them.

Comments on the issues discussed in this memo are gladly taken either privately or on the model-t mailing list (https://www.ietf.org/mailman/listinfo/Model-t).

Some further work includes items listed in Section 4.1 and Section 4.3, as well as compiling categories of vulnerabilities that need to be addressed, examples of specific attacks, and continuing the analysis of the situation and possible new remedies.

It is also necessary find suitable use cases that the IETF can address by further work in this space. A completely adversial situation is not really workable, but there are situations where some parties are trustworthy, and wish to co-operate to show to each other that this is really the case. In these situations data minimisation can be beneficial to both, attestation can provide additional trust, detection of incidents can alert the parties to action, and so on.

6. Informative References

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

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

[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, https://arxiv.org/pdf/1801.02918.pdf , 2018.

[Bloatware]
Gamba, G., Rashed, M., Razaghpanah, A., Tapiado, J., and N. Vallina, "An Analysis of Pre-installed Android Software", arXiv preprint arXiv:1905.02713 (2019) , 2019.

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

[CommandAndControl]
Botnet, ., "Creating botnet C&C server. What architecture should I use? IRC? HTTP?", Stackexchange.com question, https://security.stackexchange.com/questions/100577/creating-botnet-cc-server-what-architecture-should-i-use-irc-http , 2014.

[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, https://www.ics.uci.edu/~seal/publications/2018ICSE_Hammad.pdf , 2018.

[DeepDive]
Krebs on Security, ., "A Deep Dive on the Recent Widespread DNS Hijacking Attacks", krebsonsecurity.com blog, https://krebsonsecurity.com/2019/02/a-deep-dive-on-the-recent-widespread-dns-hijacking-attacks/ , 2019.

[GDPRAccess]
EU, ., "Right of access by the data subject", Article 15, GDPR, https://gdpr-info.eu/art-15-gdpr/ , n.d..

[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, https://www.net.in.tum.de/fileadmin/bibtex/publications/papers/schlamp_TMA_1_2015.pdf , 2015.

[Home] Nthala, N. and I. Flechais, "Rethinking home network security", European Workshop on Usable Security (EuroUSEC), https://ora.ox.ac.uk/objects/uuid:e2460f50-579b-451b-b14e-b7be2decc3e1/download_file?safe_filename=bare_conf_EuroUSEC2018.pdf&file_format=application%2Fpdf&type_of_work=Conference+item , 2018.

[I-D.arkko-arch-dedr-report]
Arkko, J. and T. Hardie, "Report from the IAB workshop on Design Expectations vs. Deployment Reality in Protocol Development", draft-arkko-arch-dedr-report-00 (work in progress), November 2019.

[I-D.arkko-arch-infrastructure-centralisation]
Arkko, J., "Centralised Architectures in Internet Infrastructure", draft-arkko-arch-infrastructure-centralisation-00 (work in progress), November 2019.

[I-D.arkko-arch-internet-threat-model]
Arkko, J., "Changes in the Internet Threat Model", draft-arkko-arch-internet-threat-model-01 (work in progress), July 2019.

[I-D.farrell-etm]
Farrell, S., "We're gonna need a bigger threat model", draft-farrell-etm-03 (work in progress), July 2019.

[I-D.iab-protocol-maintenance]
Thomson, M., "The Harmful Consequences of the Robustness Principle", draft-iab-protocol-maintenance-04 (work in progress), November 2019.

[I-D.ietf-httpbis-expect-ct]
estark@google.com, e., "Expect-CT Extension for HTTP", draft-ietf-httpbis-expect-ct-08 (work in progress), December 2018.

[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-03 (work in progress), September 2019.

[I-D.ietf-quic-transport]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-24 (work in progress), November 2019.

[I-D.ietf-rats-eat]
Mandyam, G., Lundblade, L., Ballesteros, M., and J. O'Donoghue, "The Entity Attestation Token (EAT)", draft-ietf-rats-eat-01 (work in progress), July 2019.

[I-D.ietf-teep-architecture]
Pei, M., Tschofenig, H., Thaler, D., and D. Wheeler, "Trusted Execution Environment Provisioning (TEEP) Architecture", draft-ietf-teep-architecture-05 (work in progress), December 2019.

[I-D.ietf-teep-protocol]
Tschofenig, H., Pei, M., Wheeler, D., and D. Thaler, "Trusted Execution Environment Provisioning (TEEP) Protocol", draft-ietf-teep-protocol-00 (work in progress), December 2019.

[I-D.ietf-tls-esni]
Rescorla, E., Oku, K., Sullivan, N., and C. Wood, "Encrypted Server Name Indication for TLS 1.3", draft-ietf-tls-esni-05 (work in progress), November 2019.

[I-D.ietf-tls-grease]
Benjamin, D., "Applying GREASE to TLS Extensibility", draft-ietf-tls-grease-04 (work in progress), August 2019.

[I-D.lazanski-smart-users-internet]
Lazanski, D., "An Internet for Users Again", draft-lazanski-smart-users-internet-00 (work in progress), July 2019.

[I-D.mcfadden-smart-endpoint-taxonomy-for-cless]
McFadden, M., "Endpoint Taxonomy for CLESS", draft-mcfadden-smart-endpoint-taxonomy-for-cless-00 (work in progress), July 2019.

[I-D.nottingham-for-the-users]
Nottingham, M., "The Internet is for End Users", draft-nottingham-for-the-users-09 (work in progress), July 2019.

[I-D.taddei-smart-cless-introduction]
Taddei, A., Wueest, C., Roundy, K., and D. Lazanski, "Capabilities and Limitations of an Endpoint-only Security Solution", draft-taddei-smart-cless-introduction-01 (work in progress), July 2019.

[Kocher2019]
Kocher, P., Horn, J., Fogh, A., Genkin, D., Gruss, D., Haas, W., Hamburg, M., Lipp, M., Mangard, S., Prescher, T., Schwarz, M., and Y. Yarom, "Spectre Attacks: Exploiting Speculative Execution", 40th IEEE Symposium on Security and Privacy (S&P'19) , 2019.

[LeakyBuckets]
Chickowski, E., "Leaky Buckets: 10 Worst Amazon S3 Breaches", Bitdefender blog, https://businessinsights.bitdefender.com/worst-amazon-breaches , 2018.

[Lipp2018]
Lipp, M., Schwarz, M., Gruss, D., Prescher, T., Haas, W., Fogh, A., Horn, J., Mangard, S., Kocher, P., Genkin, D., Yarom, Y., and M. Hamburg, "Meltdown: Reading Kernel Memory from User Space", 27th USENIX Security Symposium (USENIX Security 18) , 2018.

[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, https://www.degruyter.com/downloadpdf/j/popets.2018.2018.issue-1/popets-2018-0006/popets-2018-0006.pdf , 2018.

[MeltdownAndSpectre]
CISA, ., "Meltdown and Spectre Side-Channel Vulnerability Guidance", Alert (TA18-004A), https://www.us-cert.gov/ncas/alerts/TA18-004A , 2018.

[Passwords]
com, haveibeenpwned., "Pwned Passwords", Website https://haveibeenpwned.com/Passwords , 2019.

[RFC1958] Carpenter, B., Ed., "Architectural Principles of the Internet", RFC 1958, DOI 10.17487/RFC1958, June 1996, <https://www.rfc-editor.org/info/rfc1958>.

[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>.

[RFC3935] Alvestrand, H., "A Mission Statement for the IETF", BCP 95, RFC 3935, DOI 10.17487/RFC3935, October 2004, <https://www.rfc-editor.org/info/rfc3935>.

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <https://www.rfc-editor.org/info/rfc4655>.

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

[RFC6480] Lepinski, M. and S. Kent, "An Infrastructure to Support Secure Internet Routing", RFC 6480, DOI 10.17487/RFC6480, February 2012, <https://www.rfc-editor.org/info/rfc6480>.

[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>.

[RFC6797] Hodges, J., Jackson, C., and A. Barth, "HTTP Strict Transport Security (HSTS)", RFC 6797, DOI 10.17487/RFC6797, November 2012, <https://www.rfc-editor.org/info/rfc6797>.

[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>.

[RFC7469] Evans, C., Palmer, C., and R. Sleevi, "Public Key Pinning Extension for HTTP", RFC 7469, DOI 10.17487/RFC7469, April 2015, <https://www.rfc-editor.org/info/rfc7469>.

[RFC7540] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", RFC 7540, DOI 10.17487/RFC7540, May 2015, <https://www.rfc-editor.org/info/rfc7540>.

[RFC7817] Melnikov, A., "Updated Transport Layer Security (TLS) Server Identity Check Procedure for Email-Related Protocols", RFC 7817, DOI 10.17487/RFC7817, March 2016, <https://www.rfc-editor.org/info/rfc7817>.

[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>.

[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>.

[RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <https://www.rfc-editor.org/info/rfc8484>.

[RFC8546] Trammell, B. and M. Kuehlewind, "The Wire Image of a Network Protocol", RFC 8546, DOI 10.17487/RFC8546, April 2019, <https://www.rfc-editor.org/info/rfc8546>.

[RFC8555] Barnes, R., Hoffman-Andrews, J., McCarney, D., and J. Kasten, "Automatic Certificate Management Environment (ACME)", RFC 8555, DOI 10.17487/RFC8555, March 2019, <https://www.rfc-editor.org/info/rfc8555>.

[Saltzer] Saltzer, J., Reed, D., and D. Clark, "End-To-End Arguments in System Design", ACM TOCS, Vol 2, Number 4, pp 277-288 , November 1984.

[Savage] Savage, S., "Modern Automotive Vulnerabilities: Causes, Disclosures, and Outcomes", USENIX , 2016.

[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), https://www.ndss-symposium.org/wp-content/uploads/2018/06/eurousec2018_16_Malkin_paper.pdf" , 2018.

[StackEvo] Trammell, B., Thomson, M., Howard, L., and T. Hardie, "What Is an Endpoint?", Unpublished work, https://github.com/stackevo/endpoint-draft/blob/master/draft-trammell-whats-an-endpoint.md , 2017.

[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, https://conferences.sigcomm.org/sigcomm/2010/papers/sigcomm/p363.pdf , 2011.

[TargetAttack] Osborne, C., "How hackers stole millions of credit card records from Target", ZDNET, https://www.zdnet.com/article/how-hackers-stole-millions-of-credit-card-records-from-target/ , 2014.

[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, https://arxiv.org/pdf/1805.02751.pdf , 2019.

[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, https://scholarspace.manoa.hawaii.edu/bitstream/10125/50485/paper0598.pdf , 2018.

[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, https://faculty.washington.edu/kstarbi/examining-trolls-polarization.pdf , 2018.

[Unread] Obar, J. and A. Oeldorf, "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.

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

Appendix A. Acknowledgements

The authors would like to thank the IAB:

Alissa Cooper, Wes Hardaker, Ted Hardie, Christian Huitema, Zhenbin Li, Erik Nordmark, Mark Nottingham, Melinda Shore, Jeff Tantsura, Martin Thomson, Brian Trammel, Mirja Kuhlewind, and Colin Perkins.

The authors would also like to thank the participants of the IETF SAAG meeting where this topic was discussed:

Harald Alvestrand, Roman Danyliw, Daniel Kahn Gilmore, Wes Hardaker, Bret Jordan, Ben Kaduk, Dominique Lazanski, Eliot Lear, Lawrence Lundblade, Kathleen Moriarty, Kirsty Paine, Eric Rescorla, Ali Rezaki, Mohit Sethi, Ben Schwartz, Dave Thaler, Paul Turner, David Waltemire, and Jeffrey Yaskin.

The authors would also like to thank the participants of the IAB 2019 DEDR workshop:

Tuomas Aura, Vittorio Bertola, Carsten Bormann, Stephane Bortzmeyer, Alissa Cooper, Stephen Farrell, Hannu Flinck, Carl Gahnberg, Phillip Hallam-Baker, Ted Hardie, Paul Hoffman, Christian Huitema, Geoff Huston, Konstantinos Komaitis, Mirja Kuhlewind, Dirk Kutscher, Zhenbin Li, Julien Maisonneuve, John Mattson, Moritz Muller, Joerg Ott, Lucas Pardue, Jim Reid, Jan-Frederik Rieckers, Mohit Sethi, Melinda Shore, Jonne Soininen, Andrew Sullivan, and Brian Trammell.

The authors would also like to thank the participants of the November 2016 meeting at the IETF:

Carsten Bormann, Tommy C, Roman Danyliw, Christian Huitema, Ben Kaduk, Dirk Kutscher, Dominique Lazanski, Eric Rescorla, Ali Rezaki, Melinda Shore, Martin Thomson, and Robin Wilton ... (missing many people... did we have minutes other than the list of actions?) ...

Finally, the authors would like to thank numerous other people for insightful comments and discussions in this space.

Authors' Addresses

Jari Arkko
Ericsson

Email: jari.arkko@piuha.net

Stephen Farrell
Trinity College Dublin

Email: stephen.farrell@cs.tcd.ie