Internet Architecture Board (IAB)
Request for Comments: 7687
Category: Informational
ISSN: 2070-1721

S. Farrell
Trinity College, Dublin
R. Wenning
B. Bos
W3C
M. Blanchet
Viagenie
H. Tschofenig
ARM Ltd.

December 2015

Report from the Strengthening the Internet (STRINT) Workshop

Abstract

The Strengthening the Internet (STRINT) workshop assembled one hundred participants in London for two days in early 2014 to discuss how the technical community, and in particular the IETF and the W3C, should react to Pervasive Monitoring and more generally how to strengthen the Internet in the face of such attacks. The discussions covered issues of terminology, the role of user interfaces, classes of mitigation, some specific use cases, transition strategies (including opportunistic encryption), and more. The workshop ended with a few high-level recommendations, that it is believed could be implemented and could help strengthen the Internet. This is the report of that workshop.

Note that this document is a report on the proceedings of the workshop. The views and positions documented in this report are those of the workshop participants and do not necessarily reflect IAB views and positions.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Architecture Board (IAB) and represents information that the IAB has deemed valuable to provide for permanent record. Documents approved for publication by the IAB are not a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc7687.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Context

The technical plenary session at IETF 88 [vancouverplenary] concluded that Pervasive Monitoring (PM) represents an attack on the Internet, and the IETF has begun to carry out the more obvious actions required to try to handle this attack. However, there are much more complex questions arising that need further consideration before any additional concrete plans can be made.

The W3C (<https://www.w3.org>) and IAB (<https://www.iab.org>) therefore decided to host a workshop on the topic of "Strengthening the Internet Against Pervasive Monitoring" [STRINT] before IETF 89 in London in March 2014. The FP7-funded STREWS project (<https://www.strews.eu/>) organised the STRINT workshop on behalf of the IAB and W3C.

The main workshop goal was to discuss what can be done, especially by the two standards organisations IETF and W3C, against PM, both for existing Internet protocols (HTTP/1, SMTP, etc.) and for new ones (WebRTC, HTTP/2, etc.).

The starting point for the workshop was the existing IETF consensus that PM is an attack [RFC7258] (the text of which had achieved IETF consensus at the time of the workshop, even though the RFC had yet to be published).

2. Summary

The workshop was well attended (registration closed when the maximum capacity of 100 was reached, but more than 150 expressed a desire to register) and several people (about 165 at the maximum) listened to the streaming audio. The submitted papers (67 in total) were generally of good quality and all were published, except for a few where authors who couldn't take part in the workshop preferred not to publish.

The chairs of the workshop summarised the workshop in the final session in the form of the following recommendations:

1. Well-implemented cryptography can be effective against PM and will benefit the Internet if used more, despite its cost, which is steadily decreasing anyway.

2. Traffic analysis also needs to be considered, but is less well understood in the Internet community: relevant research and protocol mitigations such as data minimisation need to be better understood.

3. Work should continue on progressing the PM threat model document [Barnes] discussed in the workshop. Subsequent work on this topic resulted in the publication of [RFC7624].

4. Later, the IETF may be in a position to start to develop an update to BCP 72 [RFC3552], most likely as a new RFC enhancing that BCP and dealing with recommendations on how to mitigate PM and how to reflect that in IETF work.

5. The term "opportunistic" has been widely used to refer to a possible mitigation strategy for PM. The community needs to document definition(s) for this term, as it is being used differently by different people and in different contexts. We may also be able to develop a cookbook-like set of related protocol techniques for developers. Since the workshop, the IETF's Security area has taken up this work, most recently favouring the generic term "Opportunistic Security" (OS) [Kent]. Subsequent work on this topic resulted in the publication of a definition of OS in [RFC7435].

6. The technical community could do better in explaining the real technical downsides related to PM in terms that policy makers can understand.

7. Many user interfaces (UIs) could be better in terms of how they present security state, though this is a significantly hard problem. There may be benefits if certain dangerous choices were simply not offered anymore. But that could require significant coordination among competing software makers; otherwise, some will be considered "broken" by users.

8. Further discussion is needed on ways to better integrate UI issues into the processes of IETF and W3C.

9. Examples of good software configurations that can be cut-and-pasted for popular software, etc., can help. This is not necessarily standards work, but maybe the standards organisations can help and can work with those developing such package-specific documentation.

10. The IETF and W3C can do more so that default ("out-of-the-box") settings for protocols better protect security and privacy.

11. Captive portals [Captive] and some firewalls, too, can and should be distinguished from real man-in-the-middle attacks. This might mean establishing common conventions with makers of such middleboxes, but might also mean developing new protocols. However, the incentives for deploying such new middlebox features might not align.

3. Workshop Goals

As stated, the STRINT workshop started from the position [RFC7258] that PM is an attack. While some dissenting voices are expected and need to be heard, that was the baseline assumption for the workshop, and the high-level goal was to provide more consideration of that and how it ought to affect future work within the IETF and W3C.

At the next level down, the goals of the STRINT workshop were to:

- Discuss and hopefully come to agreement among the participants on concepts in PM for both threats and mitigation, e.g., "opportunistic" as the term applies to cryptography.

- Discuss the PM threat model, and how that might be usefully documented for the IETF at least, e.g., via an update to BCP 72. [RFC3552]

- Discuss and progress common understanding in the trade-offs between mitigating and suffering PM.
- Identify weak links in the chain of Web security architecture with respect to PM.
- Identify potential work items for the IETF, IAB, IRTF, and W3C that would help mitigate PM.
- Discuss the kinds of action outside the IETF/W3C context that might help those done within the IETF/W3C.

4. Workshop Structure

The workshop structure was designed to maximise discussion time. There were no direct presentations of submitted papers. Instead, the moderators of each session summarised topics that the Technical Programme Committee (TPC) had agreed based on the submitted papers. These summary presentations took at most 50% of the session and usually less.

Because the papers would not be presented during the workshop, participants were asked to read and discuss the papers beforehand, at least those relevant to their fields of interest. (To help people choose papers to read, authors were asked to provide short abstracts.)

Most of the sessions had two moderators, one to lead the discussion, while the other managed the queue of people who wanted to speak. This worked well: everybody got a chance to speak and each session still ended on time.

The penultimate session consisted of break-outs (which turned out to be the most productive sessions of all, most likely simply due to the smaller numbers of people involved). The subjects for the break-outs were agreed during the earlier sessions, and just before the break-out session the participants collectively determined who would attend which.

5. Topics

The following sections contain summaries of the various sessions. See the minutes (see Appendix B) for more details.

5.1. Opening session

The first session discussed the goals of the workshop. Possible approaches to improving security in the light of pervasive monitoring include a critical look at what metadata is actually required, whether old (less secure) devices can be replaced with new ones, what are "low-hanging fruit" (issues that can be handled quickly and easily), and what level of security is "good enough": a good solution may be one that is good for 90% of people or 90% of organisations.

Some participants felt that standards are needed so that people can see if their systems conform to a certain level of security, and easy to remember names are needed for those standards, so that a buyer can immediately see that a product "conforms to the named intended standard."

5.2. Threats

One difference between "traditional" attacks and pervasive monitoring is modus operandi of the attacker: typically, one determines what resources an attacker might want to target and at what cost and then one defends against that threat. But a pervasive attacker has no specific targets, other than to collect everything he can. The calculation of the cost of losing resources vs. the cost of protecting them is thus different. And unlike someone motivated to make money, a PM attacker may not be concerned at the cost of the attack (or may even prefer a higher cost, for "empire building" reasons).

The terminology used to talk about threats has to be chosen carefully (this was a common theme in several sessions), because we need to explain to people outside the technical community what they need to do or not do. For example, authentication of endpoints doesn’t so much "protect against" man-in-the-middle (MITM) attacks as make them visible. The attacker can still mount an attack but does not remain invisible while he does so. Somebody on either end of the conversation needs to react to the alert from the system: stop the conversation or find a different channel.

Paradoxically, while larger sites such as Facebook, Yahoo, and Google supervise the security of their respective services more than other smaller sites, such large sites offer a much more attractive target to attack. Avoiding overuse of such repositories for private or

sensitive information may be a useful measure that increases the cost of collecting for a pervasive attacker. This is sometimes called the target-dispersal approach.

Lack of interoperability between systems can lead to poorly thought out work-arounds and compromises that may themselves introduce vulnerabilities. Thus, improving interoperability needs to be high on the list of priorities of standards makers and even more for implementers. Of course, testing (such as interop testing) is, at some level, part of the process of the IETF and W3C; and the W3C is currently increasing its testing efforts.

5.3. Increase Usage of Security Tools

The first session on Communication Security (COMSEC) tools looked at the question why existing security tools aren't used more.

The example of the public key infrastructure used to secure HTTP is informative. One problem is that certification authorities (CAs) may issue a certificate for any domain. Thus, a single compromised CA can be used in combination with a MITM to impersonate any server. Moreover, ongoing administration, including requesting, paying for, and installing new certificates, has proven over time to be an insurmountable barrier for many web site administrators, leading them not to bother to secure their systems.

Some ideas were discussed for improving the CA system, e.g., via cross-certification of CAs and by means of "certificate transparency" -- a public, permanent log of who issued which certificate [RFC6962].

Using other models than the hierarchical certificate model (as alternative or in combination) may also help. Pretty Good Privacy (PGP) demonstrates a model known as a "web of trust" where people verify the public key of the people they meet. Because there is no innate transitive trust in PGP, it is appropriate only for small-scale uses; an example is a team of people working on a project.

Yet another model is "trust on first use" (TOFU). This is used quite effectively by SSH [RFC4252]. On the first connection, one has no way to verify that the received public key belongs to the server one is contacting, therefore, the key is accepted without further verification. But on the subsequent connections, one can verify that the received key is the same key as the first time. So, a MITM has to be there on all connections, including the first; otherwise, it will be detected by a key mismatch.

This works well for SSH, because people typically use SSH to communicate with a small number of servers over and over again. And, if they want, they may find a separate channel to get the public key (or its fingerprint). It may also work for web servers used by small groups (the server of a sports club, a department of a company, etc.), but probably works less well for public servers that are visited once or a few times or for large services where many servers may be used.

A similar proposal [RFC7469] for an HTTP header introduces an aspect of TOFU into HTTP: Key pinning tells HTTP clients that for a certain time after receiving this certificate, they should not expect the certificate to change. If it does, even if the new certificate looks valid, the client should assume a security breach.

The Session Initiation Protocol (SIP) [RFC3261] can require several different intermediaries in different stages of the communication to deal with NAT traversal and to handle policy. While both hop-by-hop and end-to-end encryption are specified, in practice, many SIP providers disable these functions. The reasons for disabling end-to-end security here are understandable: to overcome lack of interoperability they often need to change protocol headers and modify protocol data. Some workshop participants argued that SIP would never have taken off if it hadn't been possible for providers to monitor and interfere in communications in this way. Of course, that means an attacker can listen in just as easily.

A new protocol for peer-to-peer communication of video and audio (and potentially other data) is WebRTC. WebRTC reuses many of the same architectural concepts as SIP, but there is a reasonable chance that it can do better in terms of protecting users: The people implementing the protocols and offering the service have different goals and interests. In particular, the first implementers are browser makers, who may have different business models from other more traditional Voice over IP providers.

XMPP [RFC6120] suffers from yet a different kind of problem. It has encryption and authentication, and the OTR ("off the record") extension even provides what is called Perfect Forward Secrecy (PFS), i.e., compromising the current communication never gives an attacker enough information to decrypt past communications that he may have recorded. But, in practice, many people don't use XMPP at all, but rather Skype, WhatsApp, or other instant-messaging tools with unknown or no security. The problem here seems to be one of user awareness. And though OTR does provide security, it is not well integrated with XMPP, nor is it available as a core feature of XMPP clients.

To increase usage of existing solutions, some tasks can be identified; though how those map to actions for, e.g., IETF/W3C is not clear:

- Improvements to the certificate system, such as certificate transparency (CT).
- Making it easier (cheaper, quicker) for system administrators to deploy secure solutions.
- Improve awareness of the risks. Identify which communities influence which decisions and what is the appropriate message for each.
- Provide an upgrade path that doesn't break existing systems or require that everybody upgrade at the same time. Opportunistic Security may be one model for that.

5.4. Policy Issues and Non-technical Actions

Previous sessions already concluded that the problem isn't just technical, such as getting the right algorithms in the standards, fixing interoperability, or educating implementers and systems administrators. There are user interface issues and education issues too. And there are also legal issues and policy issues for governments.

It appears that the public, in general, demands more privacy and security (e.g., for their children) but are also pessimistic about getting that. They trust that somebody assures that nothing bad happens to them, but they also expect to be spied on all the time.

(Perceived) threats of terrorism gave governments a reason to allow widespread surveillance, far beyond what may previously have been considered dangerous for freedom.

In this environment, the technical community will have a hard time developing and deploying technologies that fully counter PM, which means there has to be action in the social and political spheres, too.

Technology isn't the only thing that can make life harder for attackers. Government-sponsored PM is indirectly affected by trade agreements and treaties, and thus it makes sense to lobby for those to be as privacy-friendly as possible.

Court cases on the grounds of human rights can also influence policy, especially if they reach, for example, the European Court of Human Rights.

In medicine and law, it is common to have ethics committees, not so in software. Should standards bodies such as the IETF and W3C have an ethics committee? Standards such as the Geolocation API [w3c-geo-api] have gotten scrutiny from privacy experts, but only in an ad hoc manner. (W3C has permanent groups to review standards for accessibility and internationalisation. It also has a Privacy group, but that currently doesn't do the same kind of systematic reviews.)

As the Internet-Draft draft-barnes-pervasive-problem-00 [Barnes] (which was included as paper 44) explains, PM doesn't just monitor the networks, but also attacks at the endpoints, turning organisations or people into (willing, unwilling, or unwitting) collaborators. Note: that document later evolved into [RFC7624]. One technical means of protection is thus to design protocols such that there are fewer potential collaborators, e.g., a provider of cloud storage cannot hand over plaintext for content that is encrypted with a key he doesn't have, and cannot hand over names if his client is anonymous.

It is important to distinguish between PM and fighting crime. PM is an attack, but a judge ordering the surveillance of a suspected criminal is not. The latter, often abbreviated in this context as LI (for Lawful Intercept) is outside the scope of this workshop.

5.5. Improving the Tools

An earlier session discussed why existing COMSEC tools weren't used more. This second session on COMSEC therefore discussed what improvements and/or new tools were needed.

Discussion at the workshop indicated that an important meta-tool for improving existing security technology could be Opportunistic Security (OS) [Kent]. The idea is that software is enhanced with a module that tries to encrypt communications when it detects that the other end also has the same capability, but otherwise it lets the communication continue in the old way. The detailed definition of OS was being discussed by the IETF Security Area Advisory Group at the time of this workshop [SAAG_list].

OS would protect against a passive eavesdropper but should also allow for endpoint authentication to protect against an active attacker (a MITM). As OS spreads, more and more communications would be encrypted (and hopefully authenticated), and thus there is less and less for an eavesdropper to collect.

Of course, an implementation of OS could give a false sense of security as well: some connections are encrypted, some are not. A user might see something like a padlock icon in browsers, but there was agreement at the workshop that such user interface features ought not be changed because OS is being used.

There is also the possibility that a MITM intercepts the reply from a server that says "yes, I can do encryption" and removes it, causing the client to fall back to an unencrypted protocol. Mitigations against this can be to have other channels of finding out a server's capabilities and remembering that a server could do encryption previously.

There is also, again, a terminology problem. The technical descriptions of OS talk about "silent fail" when a connection couldn't be encrypted and has to fall back to the old, unencrypted protocol. Actually, it's not a fail; it's no worse than it was before. A successful encryption would rather be a "silent improvement."

That raises the question of the UI: How do you explain to a user what their security options are, and, in case an error occurs, how do you explain the implications of the various responses?

The people working on encryption are mathematicians and engineers, and typically not the same people who know about UI. We need to involve the experts. We also need to distinguish between usability of the UI, user understanding, and user experience. For an e-commerce site, e.g., it is not just important that the user's data is technically safe, but also that he feels secure. Otherwise, he still won't buy anything.

When talking about users, we also need to distinguish the end user (who we typically think about when we talk about UI) from the server administrators and other technical people involved in enabling a connection. When something goes wrong (e.g., the user's software detects an invalid certificate), the message usually goes to the end user. But, he isn't necessarily the person who can do something about it. For example, if the problem is a certificate that expired yesterday, the options for the user are to break the connection (the safe choice, but it means he can't get his work done) or continue anyway (there could be a MITM). The server administrator, on the other hand, could actually solve the problem.

Encryption and authentication have a cost, in terms of setting them up, but also in terms of the time it takes for software to do the calculations. The setup cost can be reduced with sensible defaults, predefined profiles, and cut-and-paste configurations. And for some

connections, authentication without encryption could be enough, in the case that the data doesn't need to be kept secret, but it is important to know that it is the real data. Most mail user agents (UA) already provide independent options for encryption and signing, but web servers only support authentication if the connection is also encrypted.

On the other hand, as email also shows, it is difficult for users to understand what encryption and authentication do separately.

It also has to be kept in mind that encrypting only the "sensitive" data and not the rest decreases the cost for an attacker, too: It becomes easy to know which connections are worth attacking. Selective field confidentiality is also more prone to lead to developer error, as not all developers will know the provenance of values to be processed.

One problem with the TOFU model as used by SSH (see explanation above) is that it lacks a solution for key continuity: When a key is changed (which can happen, e.g., when a server is replaced or the software upgraded), there is no way to inform the client. (In practice, people use other means, such as calling people on the phone or asking their colleagues in the office, but that doesn't scale and doesn't always happen either.) An improvement in the SSH protocol could thus be a way to transfer a new key to a client in a safe way.

5.6. Hiding Metadata

Encryption and authentication help protect the content of messages. Correctly implemented encryption is very hard to crack. (To get the content, an attacker would rather attempt to steal the keys, corrupt the encoding software, or get the content via a collaborator. See [RFC7624] for more information on "collaborator".) But encrypting the content doesn't hide the fact that you are communicating. This metadata (who talks to whom, when, and for how long) is often as interesting as the content itself, and in some cases the size and timing of messages is even an accurate predictor of the content. So how to stop an attacker from collecting metadata, given that much of that data is actually needed by routers and other services to deliver the message to the right place?

It is useful to distinguish different kinds of metadata: explicit (or metadata proper) and implicit (sometimes called traffic data). Implicit metadata is things that can be derived from a message or are necessary for its delivery, such as the destination address, the size, the time, or the frequency with which messages pass. Explicit

metadata is things like quality ratings, provenance, or copyright data: data about the data, useful for an application, but not required to deliver the data to its endpoint.

A system such as Tor hides much of the metadata by passing through several servers, encrypting all the data except that which a particular server needs to see. Each server thus knows which server a message came from and where it has to send it to, but cannot know where the previous server got it from or where the next server is instructed to send it. However, deliberately passing through multiple servers makes the communication slower than taking the most direct route and increases the amount of traffic the network as a whole has to process.

There are three kinds of measures that can be taken to make metadata harder to get: aggregation, contraflow, and multipath (see "Flows and Pervasive Monitoring" [Paper4]). New protocols should be designed such that these measures are not inadvertently disallowed, e.g., because the design assumes that the whole of a conversation passes through the same route.

"Aggregation" means collecting conversations from multiple sources into one stream. For example, if HTTP connections pass through a proxy, all the conversations appear to come from the proxy instead of from their original sources. (This assumes that telltale information in the headers is stripped by the proxy or that the connection is encrypted.) It also works in the other direction: if multiple web sites are hosted on the same server, an attacker cannot see which of those web sites a user is reading. (This assumes that the name of the site is in the path info of the URL and not in the domain name; otherwise, watching DNS queries can still reveal the name.)

"Contraflow" means routing a conversation via one or more other servers than the normal route, e.g., by using a tunnel (e.g., with SSH or a VPN) to another server. Tor is an example of this. An attacker must watch more routes and do more effort to correlate conversations. (Again, this assumes that there is no telltale information left in the messages that leave the tunnel.)

"Multipath" splits up a single conversation (or a set of related conversations) and routes the parts in different ways, e.g., sending a request via a satellite link and receiving the response via a land line, or starting a conversation on a cellular link and continuing it via Wi-Fi. This again increases the cost for an attacker, who has to monitor and correlate data traversing multiple networks.

Protecting metadata automatically with technology at a lower layer than the application layer is difficult. The applications themselves need to pass less data, e.g., use anonymous temporary handles instead of permanent identifiers. There is often no real need for people to use the same identifier on different computers (smartphone, desktop, etc.) other than that the application they use was designed that way.

One thing that can be done relatively easily in the short term is to go through existing protocols to check what data they send that isn't really necessary. One candidate mentioned for such a study was XMPP.

"Fingerprinting" is the process of distinguishing different senders of messages based on metadata [RFC6973]: Clients can be recognised (or at least grouped) because their messages always have a combination of features that other clients' messages do not have. Reducing redundant metadata and reducing the number of optional features in a protocol reduces the variation between clients and thus makes fingerprinting harder.

Traffic analysis is a research discipline that produces sometimes surprising findings that are little known among protocol developers. Some collections of results are

- a selected bibliography on anonymity by the Free Haven Project <http://freehaven.net/anonbib/>,
- the yearly Symposium on Privacy Enhancing Technologies (PETS) <http://www.informatik.uni-trier.de/~Ley/db/conf/pet/index.html>, and
- the yearly Workshop on Privacy in the Electronic Society (WPES) <http://www.informatik.uni-trier.de/~Ley/db/conf/wpes/index.html>.

Techniques that deliberately change the timing or size of messages, such as padding, can also help reduce traffic analysis. Obviously, they make conversations slower and/or use more bandwidth, but in some cases that is not an issue, e.g., if the conversation is limited by the speed of a human user anyway. HTTP/2, for example, has a built-in padding mechanism. However, it is not easy to use these techniques well and make messages harder to recognise (as intended) rather than easier.

Different users in different contexts may have different security needs, so maybe the priority can be a user choice (if that can be done without making high-security users stand out from other users). Although many people would not understand what their choices are, some do, such as political activists or journalists.

5.7. Deployment, Intermediaries, and Middleboxes

Secure protocols have often been designed in the past for end-to-end security: Intermediaries cannot read or modify the messages. This is the model behind TLS, for example.

In practice, however, people have more or less valid reasons to insist on intermediaries: companies filtering incoming and outgoing traffic for viruses, inspecting content to give priority to certain applications, or caching content to reduce bandwidth.

In the presence of end-to-end encryption and authentication, these intermediaries have two choices: use fake certificates to impersonate the endpoints or have access to the private keys of the endpoints. The former is a MITM attack that is difficult to distinguish from a more malicious one, and the latter obviously decreases the security of the endpoints by copying supposedly confidential information and concentrating credentials in a single place.

As mentioned in Section 5.2 above, aggregation of data in a single place makes that place an attractive target. And in the case of PM, even if the data is not concentrated physically in one place, it is under control of a single legal entity that can be made into a collaborator.

The way Web communication with TLS typically works is that the client authenticates the server, but the server does not authenticate the client at the TLS layer. (If the user needs to be identified, that is mainly done at the application layer via username and password.) Thus, the presence of a MITM (middlebox) could be detected by the client (because of the incorrect certificate), but not by the server. If the client doesn't immediately close the connection (which they do not in many cases), the server may thus disclose information that the user would rather not have disclosed.

One widespread example of middleboxes is captive portals, as found on the Wi-Fi hotspots in hotels, airports, etc. Even the hotspots offering free access often intercept communications to redirect the user to a login or policy page.

When the communication they intercept is, e.g., the automatic update of your calendar program or a chat session, the redirect obviously doesn't work: these applications don't know how to display a web page. With the increasing use of applications, it may be a while before the user actually opens a browser. The flood of error messages may also have as a result that the user no longer reads the errors, allowing an actual malicious attack to go unnoticed.

Some operating systems now come with heuristics that try to recognise captive portals and either automatically login or show their login page in a separate application. (But, some hotspot providers apparently don't want automatic logins and actually reverse-engineered the heuristics to try and fool them.)

It seems some protocol is missing in this case. Captive portals shouldn't have to do MITM attacks to be noticed. A mechanism at the link layer or an extension to DHCP that tells a connecting device about the login page may help, although that still doesn't solve the problem for devices that do not have a web browser, such as voice over IP phones. HTTP response code 511 (defined in [RFC6585]) is another attempt at a partial solution. (It's partial because it can only work at the moment the user uses a browser to connect to a web site and doesn't use HTTPS.)

A practical problem with deployment of such a protocol may be that many such captive portals are very old and never updated. The hotel staff only knows how to reboot the system, and, as long as it works, the hotel has no incentive to buy a new one. As evidence of this: how many such systems require you to get a password and the ticket shows the price as zero? This is typically because the owner doesn't know how to reconfigure the hotspot, but he does know how to change the price in his cash register.

5.8. Break-out 1 - Research

Despite some requests earlier in the workshop, the research break-out did not discuss clean-slate approaches. The challenge was rather that the relationship between security research and standardisation needs improvement. Research on linkability is not yet well known in the IETF. But, the other side of the coin needs improvement too: While doing protocol design, standardisation organisations should indicate what specific problems are in need of more research.

The break-out then made a nonexhaustive list of topics that are in need of further research:

- The interaction of compression and encryption as demonstrated by the CRIME ("Compression Ratio Info-leak Made Easy") SSL/TLS vulnerability [Ristic]
- A more proactive deprecation of algorithms based on research results
- Mitigation for return-oriented programming attacks
- How to better obfuscate so-called "metadata"

- How to make the existence of traffic and their endpoints stealthy

5.9. Break-out 2 - Clients

Browsers are the first clients one thinks of when talking about encrypted connections, authentication, and certificates, but there are many others.

Other common cases of "false" alarms for MITM (after captive portals) include expired and misconfigured certificates. This is quite common in intranets, when the sysadmin hasn't bothered updating a certificate and rather tells his handful of users to just "click continue." The problem is on the one hand that users may not understand the difference between this case and the same error message when they connect to a server outside the company, and on the other hand that the incorrect certificate installed by the sysadmin is not easily distinguishable from an incorrect certificate from a MITM. The error message is almost the same, and the user may just click continue again.

One way to get rid of such certificates is if client software no longer offers the option to continue after a certificate error. That requires that all major clients (such as browsers) change their behaviour at the same time; otherwise, the first one to do so will be considered broken by users, because the others still work. Also, it requires a period in which that software gives increasingly strong warnings about the cut-off date after which the connection will fail with this certificate.

Yet another source of error messages is self-signed certificates. Such certificates are actually only errors for sites that are not expected to have them. If a message about a self-signed certificate appears when connecting to Facebook or Google, you're clearly not connected to the real Facebook or Google. But, for a personal web site, it shouldn't cause such scary warnings. There may be ways to improve the explanations in the error message and provide an easy way to verify the certificate (by email, phone, or some other channel) and trust it.

5.10. Break-out 3 - On by Default

One step in improving security is to require the relevant features (in particular, encryption and authentication) to be implemented in compliant products: The features are labelled as "MUST" in the standard rather than "MAY". This is sometimes referred to as Mandatory To Implement (MTI) and is the current practice for IETF protocols [RFC3365].

But, that may not be enough to counter PM. It may be that the features are there, but not used, because only very knowledgeable users or sysadmins turn them on. Or, it may be that implementations do not actually follow the MTI parts of specifications. Or, it may be that some security features are implemented, but interoperability for those doesn't really work. Or, even worse, it may be that protocol designers have only followed the letter of the MTI best practice and not its spirit, with the result that security features are hard to use or make deployment harder. One can thus argue that such features should be defined to be on by default.

Going further, one might argue that these features should not even be options, i.e., there should be no way to turn them off. This is sometimes called Mandatory To Use (MTU).

The questions raised at this session were for what protocols is on-by-default appropriate, and how can one explain to the developers of such protocols that it is needed?

Of course, there would be resistance to MTU security from implementers and deployments that practice deep packet inspection (DPI) and also perhaps from some governments. On the other hand, there may also be governments that outlaw protocols without proper encryption.

This break-out concluded that there could be value in attempting to document a new Best Current Practice for the IETF that moves from the current MTI position to one where security features are on by default. Some of the workshop participants expressed interest in authoring a draft for such a new BCP and progressing it through the IETF consensus process (where it would no doubt be controversial).

5.11. Break-out 4 - Measurement

There was a small break-out on the idea of measurement as a way to encourage or gamify the increased use of security mechanisms.

5.12. Break-out 5 - Opportunistic

This break-out considered the use of the term "opportunistic" as it applies to cryptographic security and attempted to progress the work towards arriving at an agreed-upon definition for use of that term, at it applies to IETF and W3C work.

While various terms had been used, with many people talking about opportunistic encryption, that usage was felt to be problematic both because it conflicted with the use of the same term in [RFC4322] and because it was being used differently in different parts of the community.

At the session, it was felt that the term "opportunistic keying" was better, but, as explained above, subsequent list discussion resulted in a move to the term "Opportunistic Security" (OS).

Aside from terminology, discussion focused on the use of Diffie-Hellman (D-H) key exchange as the preferred mechanism of OS, with fall back to cleartext if D-H doesn't succeed as a counter for passive attacks.

There was also, of course, the desire to be able to easily escalate from countering passive attacks to also handling endpoint authentication and thereby also countering MITM attacks.

Making OS visible to users was again considered to be undesirable, as users could not be expected to distinguish between cleartext, OS, and (one-sided or mutual) endpoint authentication.

Finally, it was noted that it may take some effort to establish how middleboxes might affect OS at different layers and that OS really is not suitable as the only mitigation to use for high-sensitivity sessions such as financial transactions.

5.13. Unofficial Transport/Routing Break-out

Some routing and transport Area Directors felt a little left out by all the application-layer break-outs, so they had their own brainstorm about what could be done at the transport and routing layers from which these notes resulted.

The LEDBAT [RFC6817] protocol was targeted towards a bulk-transfer service that is reordering- and delay-insensitive. Use of LEDBAT could offer the following benefits for an application:

a. Because it is reordering-insensitive, traffic can be sprayed across a large number of forwarding paths. Assuming such different paths exist, this would make it more challenging to capture and analyze a full interaction.

b. The application can vary the paths by indicating per packet a different flow. In IPv6, this can be done via different IPv6 flow labels. For IPv4, this can be done by encapsulating the IP packet into UDP and varying the UDP source port.

c. Since LEDBAT is delay-insensitive and applications using it would need to be as well, it would be possible to obfuscate the application signatures by varying the packet lengths and frequency.

d. This can also hide the transport header (for IP in UDP).

e. If the Reverse Path Forwarding (RPF) [RFC3704] check problem can be fixed, perhaps the source could be hidden; however, such fixes assume the traffic is within trusted perimeters.

f. The use of LEDBAT is orthogonal to the use of encryption and provides different benefits (harder to intercept the whole conversation, ability to obfuscate the traffic analysis), and it has different costs (longer latency, new transport protocol usage) to its users.

The idea of encrypting traffic from Customer Edge (CE) to CE as part of an L3VPN or such was also discussed. This could allow hiding of addresses, including source, and headers. From conversation with Ron Bonica, it's clear that some customers already do encryption (though without hiding the source address). So, rather than an enhancement, this is an existing mechanism for which deployment and use can be encouraged.

Finally, it was discussed whether it would be useful to have a means of communicating where and what layers are doing encryption on an application's traffic path. The initial idea of augmenting ICMP has some issues (not visible to application, ICMP packets frequently filtered) as well as potential work (determining how to trust the report of encryption). It would be interesting to understand if such communication is actually needed and what the requirements would be.

6. After the Workshop

Holding the workshop just before the IETF had the intended effect: a number of people went to both the workshop and the IETF, and they took the opportunity of being together at the IETF to continue the discussions.

IETF working groups meeting in London took the recommendations from the workshop into account. It was even the first item in the report about the IETF meeting by the IETF chair, Jari Arkko:

> Strengthening the security and privacy of the Internet continued to draw a lot of attention. The STRINT workshop organised by the IAB and W3C just before the IETF attracted 100 participants and over 60 papers. Even more people would have joined us, but there

was no space. During the IETF meeting, we continued discussing the topic at various working groups. A while ago we created the first working group specifically aimed at addressing some of the issues surrounding pervasive monitoring. The Using TLS for Applications (UTA) working group had its first meeting in London. But many other working groups also address these issues in their own work. The TCPM working group discussed a proposal to add opportunistic keying mechanisms directly onto the TCP protocol. And the DNSE BOF considered the possibility of adding confidentiality support to DNS queries. Finally, there is an ongoing effort to review old specifications to search for areas that might benefit from taking privacy and data minimisation better into account. [Arkko1]

Two papers that were written for the workshop, but not finished in time, are worth mentioning, too: One by the same Jari Arkko, titled "Privacy and Networking Functions" [Arkko2]; and one by Johan Pouwelse, "The Shadow Internet: liberation from Surveillance, Censorship and Servers" [Pouwelse].

7. Security Considerations

This document is all about security and privacy.

8. Informative References

[Arkko1] Arkko, J., "IETF-89 Summary", March 2014, <http://www.ietf.org/blog/2014/03/ietf-89-summary/>.

[Arkko2] Arkko, J., "Privacy and Networking Functions", March 2014, <http://www.arkko.com/ietf/strint/draft-arkko-strint-networking-functions.txt>.

[Barnes] Barnes, R., Schneier, B., Jennings, C., and T. Hardie, "Pervasive Attack: A Threat Model and Problem Statement", Work in Progress, draft-barnes-pervasive-problem-00, January 2014.

[Captive] Wikipedia, "Captive portal", October 2015, <https://en.wikipedia.org/w/index.php?title=Captive_portal&oldid=685621201>.

[Kent] Kent, S., "Opportunistic Security as a Countermeasure to Pervasive Monitoring", Work in Progress, draft-kent-opportunistic-security-01, April 2014.

[Paper4] Hardie, T., "Flows and Pervasive Monitoring", STRINT Workshop, 2014, <https://www.w3.org/2014/strint/papers/4.pdf>.

[Pouwelse] Pouwelse, J., "The Shadow Internet: liberation from Surveillance, Censorship and Servers", Work in Progress, draft-pouwelse-perpass-shadow-internet-00, February 2014.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <http://www.rfc-editor.org/info/rfc3261>.

[RFC3365] Schiller, J., "Strong Security Requirements for Internet Engineering Task Force Standard Protocols", BCP 61, RFC 3365, DOI 10.17487/RFC3365, August 2002, <http://www.rfc-editor.org/info/rfc3365>.

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

[RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, DOI 10.17487/RFC3704, March 2004, <http://www.rfc-editor.org/info/rfc3704>.

[RFC4252] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Authentication Protocol", RFC 4252, DOI 10.17487/RFC4252, January 2006, <http://www.rfc-editor.org/info/rfc4252>.

[RFC4322] Richardson, M. and D. Redelmeier, "Opportunistic Encryption using the Internet Key Exchange (IKE)", RFC 4322, DOI 10.17487/RFC4322, December 2005, <http://www.rfc-editor.org/info/rfc4322>.

[RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, DOI 10.17487/RFC6120, March 2011, <http://www.rfc-editor.org/info/rfc6120>.

[RFC6585] Nottingham, M. and R. Fielding, "Additional HTTP Status Codes", RFC 6585, DOI 10.17487/RFC6585, April 2012, <http://www.rfc-editor.org/info/rfc6585>.

[RFC6817] Shalunov, S., Hazel, G., Iyengar, J., and M. Kuehlewind, "Low Extra Delay Background Transport (LEDBAT)", RFC 6817, DOI 10.17487/RFC6817, December 2012, <http://www.rfc-editor.org/info/rfc6817>.

[RFC6962] Laurie, B., Langley, A., and E. Kasper, "Certificate Transparency", RFC 6962, DOI 10.17487/RFC6962, June 2013, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc7258>.

[RFC7435] Dukhovni, V., "Opportunistic Security: Some Protection Most of the Time", RFC 7435, DOI 10.17487/RFC7435, December 2014, <http://www.rfc-editor.org/info/rfc7435>.

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

[RFC7624] Barnes, R., Schneier, B., Jennings, C., Hardie, T., Trammell, B., Huitema, C., and D. Borkmann, "Confidentiality in the Face of Pervasive Surveillance: A Threat Model and Problem Statement", RFC 7624, DOI 10.17487/RFC7624, August 2015, <http://www.rfc-editor.org/info/rfc7624>.

[Ristic] Ristic, I., "CRIME: Information Leakage Attack against SSL/TLS", Qualys Blog, <https://community.qualys.com/blogs/securitylabs/2012/09/14/crime-information-leakage-attack-against-ssltls>.

[SAAG_list] IETF, "saag Discussion Archive", <https://www.ietf.org/mail-archive/web/saag/current/maillist.html>.

[STRINT] W3C/IAB, "STRINT Workshop", <https://www.w3.org/2014/strint/Overview.html>.

[vancouverplenary]
IETF, "IETF 88 Technical Plenary Minutes",
<https://www.ietf.org/proceedings/88/minutes/minutes-88-iab-techplenary>.

[w3c-geo-api]
Popescu, A., "Geolocation API Specification",
W3C Recommendation, October 2013,
<http://www.w3.org/TR/geolocation-API/>.

Appendix A. Logistics

The workshop was organised by the STREWS project (<https://www.strews.eu/>), which is a research project funded under the European Union's 7th Framework Programme (<http://cordis.europa.eu/fp7/ict/>). It was the first of two workshops in its work plan. The organisers were supported by the IAB and W3C, and, for the local organisation, by Telefonica Digital (<http://blog.digital.telefonica.com/>).

One of the suggestions in the project description of the STREWS project was to attach the first workshop to an IETF meeting. The best opportunity was IETF 89 in London, which began on Sunday 2 March 2014; see <https://www.ietf.org/meeting/89/> for more information. Telefonica Digital offered meeting rooms at its offices in central London for the preceding Friday and Saturday, just minutes away from the IETF's location.

The room held 100 people, which was thought to be sufficient. There turned out to be more interest than expected and we could have filled a larger room, but 100 people is probably an upper limit for good discussions anyway.

Apart from the usual equipment in the room (projector, white boards, microphones, coffee), we also set up some extra communication channels:

- A mailing list where participants could discuss the agenda and the published papers about three weeks in advance of the workshop itself.

- Publicly advertised streaming audio (one-way only). At some point, no less than 165 people were listening.

- An IRC channel for live minute-taking, passing links and other information, and helping remote participants to follow the proceedings.

- An Etherpad, where the authors of papers could provide an abstract of their submissions, to help participants who could not read all 66 papers in full in advance of the workshop. The abstracts were also used on the workshop's web site: <https://www.w3.org/2014/strint/>.

- A Twitter hashtag (#strint). Four weeks after the workshop, there were still a few new messages about events related to workshop topics; see <https://twitter.com/search?q=%23strint>.

Appendix B. Agenda

This was the final agenda of the workshop, as determined by the TPC and participants on the mailing list prior to the workshop. The included links are to the slides that the moderators used to introduce each discussion topic and to the minutes.

B.1. Friday 28 February

Minutes: <http://www.w3.org/2014/02/28-strint-minutes.html>

Workshop starts, welcome, logistics, opening/overview
Slides: <https://down.dsg.cs.tcd.ie/strint-slides/s0-welcome.pdf>

- Goal is to plan how we respond to PM threats
- Specific questions to be discussed in sessions
- Outcomes are actions for IETF, W3C, IRTF, etc.

I. Threats - What problem are we trying to solve?
(Presenter: Richard Barnes; Moderator: Cullen Jennings)
Slides:
<https://down.dsg.cs.tcd.ie/strint-slides/s1-threat.pdf>

* What attacks have been described? (Attack taxonomy)
* What should we assume the attackers' capabilities are?
* When is it really "pervasive monitoring" and when is it not?
* Scoping - what's in and what's out? (for IETF/W3C)

II. COMSEC 1 - How can we increase usage of current COMSEC tools?
(Presenter: Hannes Tschofenig; Moderator: Leif Johansson)
Slides:
<https://down.dsg.cs.tcd.ie/strint-slides/s2-comsec.pdf>

* Whirlwind catalog of current tools
* Why aren't people using them? In what situations are / aren't they used?
* Securing AAA and management protocols - why not?
* How can we (IETF/W3C/community) encourage more/better use?

III. Policy - What policy/legal/other issues need to be taken into account? (Presenter: Christine Runnegar; Moderator: Rigo Wenning)
Slides:
<https://down.dsg.cs.tcd.ie/strint-slides/s3-policy.pdf>

* What non-technical activities do we need to be aware of?

* How might such non-technical activities impact on IETF/W3C?

* How might IETF/W3C activities impact those non-technical activities?

Saturday plan, open mic, wrap-up of the day

B.2. Saturday 1 March

Minutes: <http://www.w3.org/2014/03/01-strint-minutes.html>

IV. COMSEC 2 - What improvements to COMSEC tools are needed? (Presenter: Mark Nottingham; Moderator: Steve Bellovin)
Slides:
<https://down.dsg.cs.tcd.ie/strint-slides/s4-opportunistic.pdf>

* Opportunistic encryption - what is it and where it might apply

* Mitigations aiming to block PM vs. detect PM - when to try which?

V. Metadata - How can we reduce the metadata that protocols expose? (Presenters: Alfredo Pironti, Ted Hardie; Moderator: Alissa Cooper)

Slides:
<https://down.dsg.cs.tcd.ie/strint-slides/s5-1metadata-pironti.pdf>
<https://down.dsg.cs.tcd.ie/strint-slides/s5-2metadata-hardie.pdf>
<https://down.dsg.cs.tcd.ie/strint-slides/s5-3metadata-cooper.pdf>

* Metadata, fingerprinting, minimisation

* What’s out there?

* How can we do better?

VI. Deployment - How can we address PM in deployment / operations? (Presenter: Eliot Lear; Moderator: Barry Leiba)
Slides: <https://down.dsg.cs.tcd.ie/strint-slides/s6-deploy.pdf>

* "Mega"-commercial services (clouds, large-scale email and Online Social Networks, SIP, WebRTC)

* Target dispersal - good goal or wishful thinking?

* Middleboxes: when a help and when a hindrance?

VII. Break-out Sessions (x 3) / Bar-Camp style (Hannes Tschofenig)

* Content to be defined during meeting, as topics come up

* Sum up at the end to gather conclusions for report

Break-outs:

1. Research Questions (Moderator: Kenny Paterson)

    + Do we need more/different crypto tools?

    + How can applications make better use of COMSEC tools?

    + What research topics could be handled in IRTF?

    + What other research would help?

2. Clients

3. On by default

4. Measurement

5. Opportunistic

VIII. Break-out Reports, Open Mic & Conclusions - What are we going to do to address PM?
Slides: <https://www.w3.org/2014/strint/slides/summary.pdf>

* Gather conclusions / recommendations / goals from earlier sessions

Appendix C. Workshop Chairs and Program Committee

The workshop chairs were three: Stephen Farrell (TCD) and Rigo Wenning (W3C) from the STREWS project, and Hannes Tschofenig (ARM) from the STREWS Interest Group.

The Technical Programme Committee (TPC) was charged with evaluating the submitted papers. It was made up of the members of the STREWS project, the members of the STREWS Interest Group, plus invited experts: Bernard Aboba (Microsoft), Dan Appelquist (Telefonica & W3C TAG), Richard Barnes (Mozilla), Bert Bos (W3C), Lieven Desmet (KU Leuven), Karen O'Donoghue (ISOC), Russ Housley (Vigil Security), Martin Johns (SAP), Ben Laurie (Google), Eliot Lear (Cisco), Kenny Paterson (Royal Holloway), Eric Rescorla (RTFM), Wendy Seltzer (W3C), Dave Thaler (Microsoft), and Sean Turner (IECA).

Appendix D. Participants

The participants to the workshop were:

- Bernard Aboba (Microsoft Corporation)
- Thijs Alkemade (Adium)
- Daniel Appelquist (Telefonica Digital)
- Jari Arkko (Ericsson)
- Alia Atlas (Juniper Networks)
- Emmanuel Baccelli (INRIA)
- Mary Barnes
- Richard Barnes (Mozilla)
- Steve Bellovin (Columbia University)
- Andrea Bittau (Stanford University)
- Marc Blanchet (Viagenie)
- Carsten Bormann (Uni Bremen TZI)
- Bert Bos (W3C)
- Ian Brown (Oxford University)
- Stewart Bryant (Cisco Systems)
- Randy Bush (IIJ / Dragon Research Labs)
- Kelsey Cairns (Washington State University)
- Stuart Cheshire (Apple)
- Vincent Cheval (University of Birmingham)
- Benoit Claise (Cisco)
- Alissa Cooper (Cisco)
- Dave Crocker (Brandenburg InternetWorking)
- Leslie Daigle (Internet Society)
- George Danezis (University College London)
- Spencer Dawkins (Huawei)
- Mark Donnelly (Painless Security)
- Nick Doty (W3C)
- Dan Druta (AT&T)

- Peter Eckersley (Electronic Frontier Foundation)
- Lars Eggert (NetApp)
- Kai Engert (Red Hat)
- Monika Ermert
- Stephen Farrell (Trinity College Dublin)
- Barbara Fraser (Cisco)
- Virginie Galindo (gemalto)
- Stefanie Gerdes (Uni Bremen TZI)
- Daniel Kahn Gillmor (ACLU)
- Wendy M. Grossman
- Christian Grothoff (The GNUnet Project)
- Oliver Hahm (INRIA)
- Joseph Lorenzo Hall (Center for Democracy & Technology)
- Phillip Hallam-Baker
- Harry Halpin (W3C/MIT and IRI)
- Ted Hardie (Google)
- Joe Hildebrand (Cisco Systems)
- Russ Housley (Vigil Security, LLC)
- Cullen Jennings (CISCO)
- Leif Johansson (SUNET)
- Harold Johnson (Irdeto)
- Alan Johnston (Avaya)
- L. Aaron Kaplan (CERT.at)
- Steve Kent (BBN Technologies)
- Achim Klabunde (European Data Protection Supervisor)
- Hans Kuhn (NOC)
- Christian de Larrinaga
- Ben Laurie (Google)
- Eliot Lear (Cisco Ssytems)
- Barry Leiba (Huawei Technologies)
- Sebastian Lekies (SAP AG)
- Orit Levin (Microsoft Corporation)
- Carlo Von LynX (#youbroketheinternet)
- Xavier Marjou (Orange)
- Larry Masinter (Adobe)
- John Mattsson (Ericsson)
- Patrick McManus (Mozilla)
- Doug Montgomery (NIST)
- Kathleen Moriarty (EMC)
- Alec Muffett (Facebook)
- Suhas Nandakumar (Cisco Systems)
- Linh Nguyen (ERCIM/W3C)
- Linus Nordberg (NORDUnet)
- Mark Nottingham
- Karen O'Donoghue (Internet Society)
- Piers O'Hanlon (Oxford Internet Institute)
- Kenny Paterson (Royal Holloway, University of London)
- Jon Peterson (Neustar)

- Joshua Phillips (University of Birmingham)
- Alfredo Pironti (INRIA)
- Dana Polatin-Reuben (University of Oxford)
- Prof. Johan Pouwelse (Delft University of Technology)
- Max Pritikin (Cisco)
- Eric Rescorla (Mozilla)
- Pete Resnick (Qualcomm Technologies, Inc.)
- Tom Ristenpart (University of Wisconsin)
- Andrei Robachevsky (Internet Society)
- David Rogers (Copper Horse)
- Scott Rose (NIST)
- Christine Runnegar (Internet Society)
- Philippe De Ryck (DistriNet - KU Leuven)
- Peter Saint-Andre (&yet)
- Runa A. Sandvik (Center for Democracy and Technology)
- Jakob Schlyter
- Dr. Jan Seedorf (NEC Laboratories Europe)
- Wendy Seltzer (W3C)
- Melinda Shore (No Mountain Software)
- Dave Thaler (Microsoft)
- Brian Trammell (ETH Zurich)
- Hannes Tschofenig (ARM Limited)
- Sean Turner (IECA, Inc.)
- Matthias Waehlisch (Freie Universitaet Berlin)
- Greg Walton (Oxford University)
- Rigo Wenning (W3C)
- Tara Whalen (Apple Inc.)
- Greg Wood (Internet Society)
- Jiangshan Yu (University of Birmingham)
- Aaron Zauner
- Dacheng Zhang (Huawei)
- Phil Zimmermann (Silent Circle LLC)
- Juan-Carlos Zuniga (InterDigital)

Authors' Addresses

Stephen Farrell
Trinity College, Dublin
Email: stephen.farrell@cs.tcd.ie
URI: https://www.cs.tcd.ie/Stephen.Farrell/

Rigo Wenning
World Wide Web Consortium
2004, route des Lucioles
B.P. 93
Sophia-Antipolis 06902
France
Email: rigo@w3.org
URI: http://www.w3.org/People/Rigo/

Bert Bos
World Wide Web Consortium
2004, route des Lucioles
B.P. 93
Sophia-Antipolis 06902
France
Email: bert@w3.org

Marc Blanchet
Viagenie
246 Aberdeen
Quebec, QC G1R 2E1
Canada
Email: Marc.Blanchet@viagenie.ca
URI: http://viagenie.ca

Hannes Tschofenig
ARM Ltd.
110 Fulbourn Rd
Cambridge CB1 9NJ
Great Britain
Email: Hannes.tschofenig@gmx.net
URI: http://www.tschofenig.priv.at