

SUIT
Internet-Draft
Intended status: Informational
Expires: January 3, 2019

B. Moran
Arm Limited
M. Meriac
Consultant
H. Tschofenig
Arm Limited
D. Brown
Linaro
July 02, 2018

A Firmware Update Architecture for Internet of Things Devices
draft-ietf-suit-architecture-01

Abstract

Vulnerabilities with Internet of Things (IoT) devices have raised the need for a solid and secure firmware update mechanism that is also suitable for constrained devices. Incorporating such update mechanism to fix vulnerabilities, to update configuration settings as well as adding new functionality is recommended by security experts.

This document lists requirements and describes an architecture for a firmware update mechanism suitable for IoT devices. The architecture is agnostic to the transport of the firmware images and associated meta-data.

This version of the document assumes asymmetric cryptography and a public key infrastructure. Future versions may also describe a symmetric key approach for very constrained devices.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 3, 2019.

Copyright Notice

Copyright (c) 2018 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. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	3
2. Conventions and Terminology	3
3. Requirements	6
3.1. Agnostic to how firmware images are distributed	6
3.2. Friendly to broadcast delivery	6
3.3. Use state-of-the-art security mechanisms	7
3.4. Rollback attacks must be prevented	7
3.5. High reliability	7
3.6. Operate with a small bootloader	8
3.7. Small Parsers	8
3.8. Minimal impact on existing firmware formats	8
3.9. Robust permissions	8
3.10. Operating modes	9
4. Claims	11
5. Communication Architecture	11
6. Manifest	14
7. Device Firmware Update Examples	15
7.1. Single CPU SoC	16
7.2. Single CPU with Secure - Normal Mode Partitioning	16

7.3. Dual CPU, shared memory	16
7.4. Dual CPU, other bus	16
8. Example Flow	17
9. IANA Considerations	18
10. Security Considerations	18
11. Mailing List Information	19
12. Acknowledgements	20
13. References	21
13.1. Normative References	21
13.2. Informative References	21
13.3. URIs	21
Authors' Addresses	22

1. Introduction

When developing IoT devices, one of the most difficult problems to solve is how to update the firmware on the device. Once the device is deployed, firmware updates play a critical part in its lifetime, particularly when devices have a long lifetime, are deployed in remote or inaccessible areas or where manual intervention is cost prohibitive or otherwise difficult. The need for a firmware update may be to fix bugs in software, to add new functionality, or to re-configure the device.

The firmware update process, among other goals, has to ensure that

- The firmware image is authenticated and attempts to flash a malicious firmware image are prevented.
- The firmware image can be confidentiality protected so that attempts by an adversary to recover the plaintext binary can be prevented. Obtaining the plaintext binary is often one of the first steps for an attack to mount an attack.

2. Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

This document uses the following terms:

- Manifest: The manifest contains meta-data about the firmware image. The manifest is protected against modification and provides information about the author.

- **Firmware Image:** The firmware image is a binary that may contain the complete software of a device or a subset of it. The firmware image may consist of multiple images, if the device contains more than one microcontroller. The image may consist of a differential update for performance reasons. Firmware is the more universal term. Both terms are used in this document and are interchangeable.
- **Bootloader:** A bootloader is a piece of software that is executed once a microcontroller has been reset. It is responsible for deciding whether to boot a firmware image that is present or whether to obtain and verify a new firmware image. Since the bootloader is a security critical component its functionality may be split into separate stages. Such a multi-stage bootloader may offer very basic functionality in the first stage and resides in ROM whereas the second stage may implement more complex functionality and resides in flash memory so that it can be updated in the future (in case bugs have been found). The exact split of components into the different stages, the number of firmware images stored by an IoT device, and the detailed functionality varies throughout different implementations.

The following entities are used:

- **Author:** The author is the entity that creates the firmware image. There may be multiple authors in a system either when a device consists of multiple micro-controllers or when the the final firmware image consists of software components from multiple companies.
- **Device:** The device is the recipient of the firmware image and the manifest. The goal is to update the firmware of the device. A single device may need to obtain more than one firmware image and manifest to successfully perform an update.
- **Communicator:** The communicator component of the device interacts with the firmware update server. It receives firmware images and triggers an update, if needed. The communicator either polls a firmware update server for the most recent manifest/firmware or manifests/firmware images are pushed to it. Note that the firmware update process may involve multiple stages since one or multiple manifests may need to be downloaded before the communicator can fetch one or multiple firmware images/software components.
- **Status Tracker:** The status tracker offers device management functionality that includes keep track of the firmware update process. This includes fine-grained monitoring of changes at the

device, for example, what state of the firmware update cycle the device is currently in.

- Firmware Server: Entity that stores firmware images and manifests. Some deployments may require storage of the firmware images/ manifests on more than one entities before they reach the device.
- Device Operator: The actor responsible for the day-to-day operation of a fleet of IoT devices.
- Network Operator: The actor responsible for the operation of a network to which IoT devices connect.

In addition to the entities in the list above there is an orthogonal infrastructure with a Trust Provisioning Authority (TPA) distributing trust anchors and authorization permissions to various entities in the system. The TPA may also delegate rights to install, update, enhance, or delete trust anchors and authorization permissions to other parties in the system. This infrastructure overlaps the communication architecture and different deployments may empower certain entities while other deployments may not. For example, in some cases, the Original Design Manufacturer (ODM), which is a company that designs and manufactures a product, may act as a TPA and may decide to remain in full control over the firmware update process of their products.

The terms 'trust anchor' and 'trust anchor store' are defined in [RFC6024]:

- "A trust anchor represents an authoritative entity via a public key and associated data. The public key is used to verify digital signatures, and the associated data is used to constrain the types of information for which the trust anchor is authoritative."
- "A trust anchor store is a set of one or more trust anchors stored in a device. A device may have more than one trust anchor store, each of which may be used by one or more applications."

Furthermore, the following abbreviations are used in this document:

- Microcontroller (MCU for microcontroller unit) is a small computer on a single integrated circuit, which is often used for mass volume IoT devices.
- System on Chip (SoC) is an integrated circuit that integrates all components of a computer, such as CPU, memory, input/output ports, secondary storage, etc.

- Homogeneous Storage Architecture (HoSA): A device that stores all firmware components in the same way, for example in a file system or in flash memory.
- Heterogeneous Storage Architecture (HeSA): A device that stores at least one firmware component differently from the rest, for example a device with an external, updatable radio, or a device with internal and external flash memory.

3. Requirements

The firmware update mechanism described in this specification was designed with the following requirements in mind:

- Agnostic to how firmware images are distributed
- Friendly to broadcast delivery
- Use state-of-the-art security mechanisms
- Rollback attacks must be prevented
- High reliability
- Operate with a small bootloader
- Small Parsers
- Minimal impact on existing firmware formats
- Robust permissions
- Diverse modes of operation

3.1. Agnostic to how firmware images are distributed

Firmware images can be conveyed to devices in a variety of ways, including USB, UART, WiFi, BLE, low-power WAN technologies, etc. and use different protocols (e.g., CoAP, HTTP). The specified mechanism needs to be agnostic to the distribution of the firmware images and manifests.

3.2. Friendly to broadcast delivery

This architecture does not specify any specific broadcast protocol however, given that broadcast may be desirable for some networks, updates must cause the least disruption possible both in metadata and payload transmission.

For an update to be broadcast friendly, it cannot rely on link layer, network layer, or transport layer security. In addition, the same message must be deliverable to many devices, both those to which it applies and those to which it does not, without a chance that the wrong device will accept the update. Considerations that apply to network broadcasts apply equally to the use of third-party content distribution networks for payload distribution.

3.3. Use state-of-the-art security mechanisms

End-to-end security between the author and the device, as shown in Section 5, is used to ensure that the device can verify firmware images and manifests produced by authorized authors.

The use of post-quantum secure signature mechanisms, such as hash-based signatures, should be explored. A migration to post-quantum secure signatures would require significant effort, therefore, mandatory-to-implement support for post-quantum secure signatures is a goal.

A mandatory-to-implement set of algorithms has to be defined offering a key length of 112-bit symmetric key or security or more, as outlined in Section 20 of RFC 7925 [RFC7925]. This corresponds to a 233 bit ECC key or a 2048 bit RSA key.

If the firmware image is to be encrypted, it must be done in such a way that every intended recipient can decrypt it. The information that is encrypted individually for each device must be an absolute minimum, for example AES Key Wrap [RFC5649], in order to maintain friendliness to Content Distribution Networks, bulk storage, and broadcast protocols.

3.4. Rollback attacks must be prevented

A device presented with an old, but valid manifest and firmware must not be tricked into installing such firmware since a vulnerability in the old firmware image may allow an attacker to gain control of the device.

3.5. High reliability

A power failure at any time must not cause a failure of the device. A failure to validate any part of an update must not cause a failure of the device. One way to achieve this functionality is to provide a minimum of two storage locations for firmware and one bootable location for firmware. An alternative approach is to use a 2nd stage bootloader with build-in full featured firmware update functionality

such that it is possible to return to the update process after power down.

Note: This is an implementation requirement rather than a requirement on the manifest format.

3.6. Operate with a small bootloader

The bootloader must be minimal, containing only flash support, cryptographic primitives and optionally a recovery mechanism. The recovery mechanism is used in case the update process failed and may include support for firmware updates over serial, USB or even a limited version of wireless connectivity standard like a limited Bluetooth Smart. Such a recovery mechanism must provide security at least at the same level as the full featured firmware update functionalities.

The bootloader needs to verify the received manifest and to install the bootable firmware image. The bootloader should not require updating since a failed update poses a risk in reliability. If more functionality is required in the bootloader, it must use a two-stage bootloader, with the first stage comprising the functionality defined above.

All information necessary for a device to make a decision about the installation of a firmware update must fit into the available RAM of a constrained IoT device. This prevents flash write exhaustion.

Note: This is an implementation requirement.

3.7. Small Parsers

Since parsers are known sources of bugs they must be minimal. Additionally, it must be easy to parse only those fields that are required to validate at least one signature or MAC with minimal exposure.

3.8. Minimal impact on existing firmware formats

The design of the firmware update mechanism must not require changes to existing firmware formats.

3.9. Robust permissions

When a device obtains a monolithic firmware image from a single author without any additional approval steps then the authorization flow is relatively simple. There are, however, other cases where

more complex policy decisions need to be made before updating a device.

In this architecture the authorization policy is separated from the underlying communication architecture. This is accomplished by separating the entities from their permissions. For example, an author may not have the authority to install a firmware image on a device in critical infrastructure without the authorization of a device operator. In this case, the device may be programmed to reject firmware updates unless they are signed both by the firmware author and by the device operator.

Alternatively, a device may trust precisely one entity, which does all permission management and coordination. This entity allows the device to offload complex permissions calculations for the device.

3.10. Operating modes

There are three broad classifications of update operating modes.

- Client-initiated Update
- Server-initiated Update
- Hybrid Update

Client-initiated updates take the form of a communicator on a device proactively checking for new firmware images provided by firmware servers.

Server-initiated updates are important to consider because timing of updates may need to be tightly controlled in some high-reliability environments. In this case the communicator, potentially in coordination with the status tracker, determines what devices qualify for a firmware update. Once those devices have been selected the firmware server distributes updates to those devices.

Note: This assumes that the firmware server is able to reach the device, which may require devices to keep reachability information at the communicator and / or at the firmware server up-to-date. This may also require keeping state at NATs and stateful packet filtering firewalls alive.

Hybrid updates are those that require an interaction between the device and the firmware server / communicator. The communicator pushes notifications of availability of an update to the device, and the device then downloads the image from the firmware server when it wants.

An alternative approach is to consider the steps a device has to go through in the course of an update:

- Notification
- Pre-authorisation
- Dependency resolution
- Download
- Installation

The notification step consists of the communicator informing the device that an update is available. This can be accomplished via polling (client-initiated), push notifications (server-initiated), or more complex mechanisms.

The pre-authorisation step involves verifying whether the entity signing the manifest is indeed authorized to perform an update. The device must also determine whether it should fetch and processing of the firmware image (unless it has been attached already to the manifest itself).

A dependency resolution phase is needed when more than one component can be updated or when a differential update is used. The necessary dependencies must be available prior to installation.

The download step is the process of acquiring a local copy of the firmware image. When the download is client-initiated, this means that the device chooses when a download occurs and initiates the download process. When a download is server-party initiated, this means that either the communicator / firmware server tells the device when to download or that it initiates the transfer directly to the device. For example, a download from an HTTP-based firmware server is client-initiated. A transfer to a LwM2M Firmware Update resource [LwM2M] is server-initiated.

If the Device has downloaded a new firmware image and is ready to install it it may need to wait for a trigger from a Communicator to install the firmware update, may trigger the update automatically, or may go through a more complex decision making process to determine the appropriate timing for an update (such as delaying the update process to a later time when end users are less impacted by the update process).

Installation is the act of processing the payload into a format that the IoT device can recognise and the bootloader is responsible for then booting from the newly installed firmware image.

Each of these steps may require different permissions.

4. Claims

Claims in the manifest offer a way to convey instructions to a device that impact the firmware update process. To have any value the manifest containing those claims must be authenticated and integrity protected. The credential used to must be directly or indirectly related to the trust anchor installed at the device by the Trust Provisioning Authority.

The baseline claims for all manifests are described in [I-D.ietf-suit-information-model]. For example, there are:

- Do not install firmware with earlier metadata than the current metadata.
- Only install firmware with a matching vendor, model, hardware revision, software version, etc.
- Only install firmware that is before its best-before timestamp.
- Only allow a firmware installation if dependencies have been met.
- Choose the mechanism to install the firmware, based on the type of firmware it is.

5. Communication Architecture

Figure 1 shows the communication architecture where a firmware image is created by an author, and uploaded to a firmware server. The firmware image/manifest is distributed to the device either in a push or pull manner using the communicator residing on the device. The device operator keeps track of the process using the status tracker. This allows the device operator to know and control what devices have received an update and which of them are still pending an update.

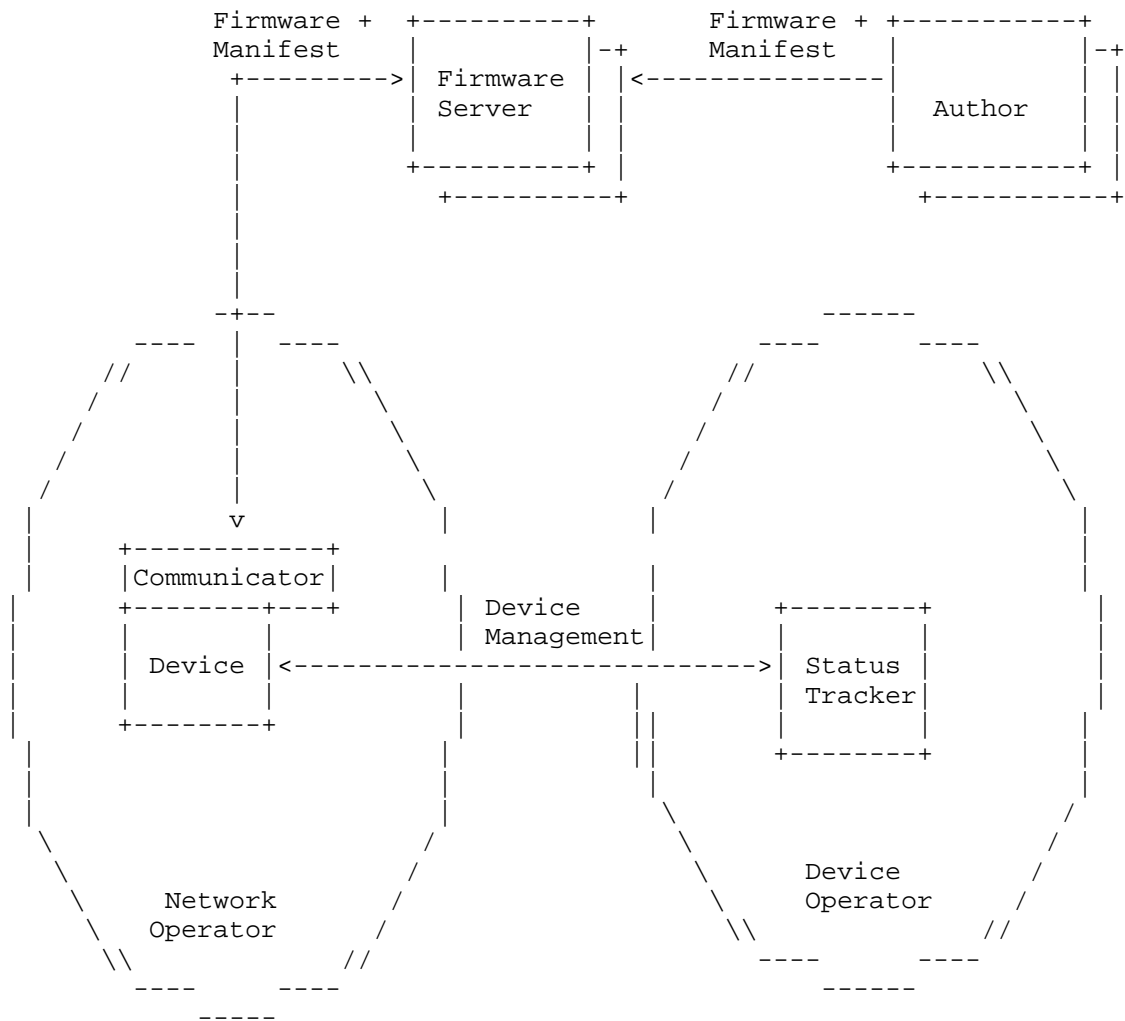


Figure 1: Architecture.

End-to-end security mechanisms are used to protect the firmware image and the manifest although Figure 2 does not show the manifest itself since it may be distributed independently.

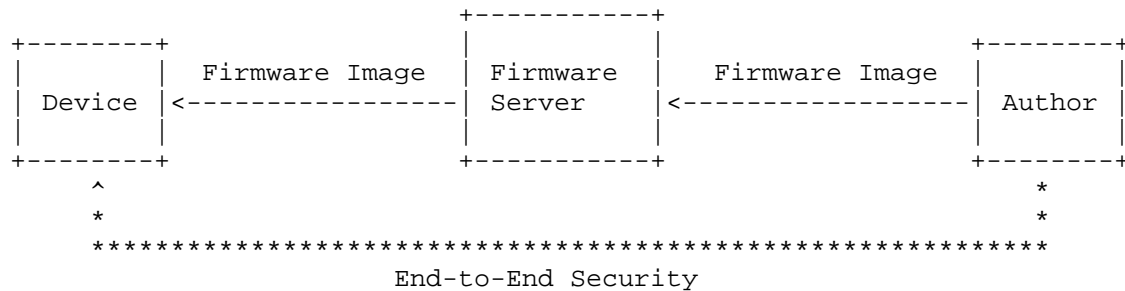


Figure 2: End-to-End Security.

Whether the firmware image and the manifest is pushed to the device or fetched by the device is a deployment specific decision.

The following assumptions are made to allow the device to verify the received firmware image and manifest before updating software:

- To accept an update, a device needs to verify the signature covering the manifest. There may be one or multiple manifests that need to be validated, potentially signed by different parties. The device needs to be in possession of the trust anchors to verify those signatures. Installing trust anchors to devices via the Trust Provisioning Authority happens in an out-of-band fashion prior to the firmware update process.
- Not all entities creating and signing manifests have the same permissions. A device needs to determine whether the requested action is indeed covered by the permission of the party that signed the manifest. Informing the device about the permissions of the different parties also happens in an out-of-band fashion and is also a duty of the Trust Provisioning Authority.
- For confidentiality protection of firmware images the author needs to be in possession of the certificate/public key or a pre-shared key of a device. The use of confidentiality protection of firmware images is deployment specific.

There are different types of delivery modes, which are illustrated based on examples below.

There is an option for embedding a firmware image into a manifest. This is a useful approach for deployments where devices are not connected to the Internet and cannot contact a dedicated server for download of the firmware. It is also applicable when the firmware update happens via a USB stick or via Bluetooth Smart. Figure 3 shows this delivery mode graphically.

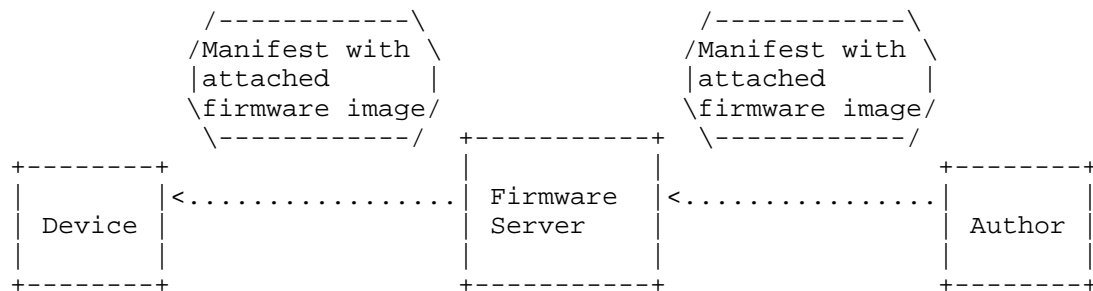


Figure 3: Manifest with attached firmware.

Figure 4 shows an option for remotely updating a device where the device fetches the firmware image from some file server. The manifest itself is delivered independently and provides information about the firmware image(s) to download.

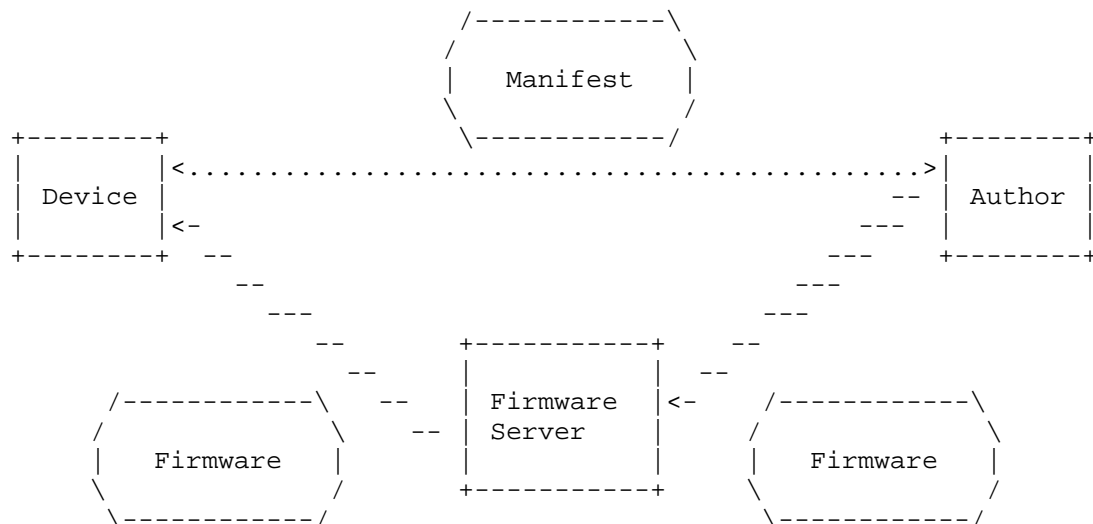


Figure 4: Independent retrieval of the firmware image.

This architecture does not mandate a specific delivery mode but a solution must support both types.

6. Manifest

In order for a device to apply an update, it has to make several decisions about the update:

- Does it trust the author of the update?

- Has the firmware been corrupted?
- Does the firmware update apply to this device?
- Is the update older than the active firmware?
- When should the device apply the update?
- How should the device apply the update?
- What kind of firmware binary is it?
- Where should the update be obtained?
- Where should the firmware be stored?

The manifest encodes the information that devices need in order to make these decisions. It is a data structure that contains the following information:

- information about the device(s) the firmware image is intended to be applied to,
- information about when the firmware update has to be applied,
- information about when the manifest was created,
- dependencies on other manifests,
- pointers to the firmware image and information about the format,
- information about where to store the firmware image,
- cryptographic information, such as digital signatures or message authentication codes (MACs).

The manifest information model is described in [I-D.ietf-suit-information-model].

7. Device Firmware Update Examples

Although these documents attempt to define a firmware update architecture that is applicable to both existing systems, as well as yet-to-be-conceived systems; it is still helpful to consider existing architectures.

7.1. Single CPU SoC

The simplest, and currently most common, architecture consists of a single MCU along with its own peripherals. These SoCs generally contain some amount of flash memory for code and fixed data, as well as RAM for working storage. These systems either have a single firmware image, or an immutable bootloader that runs a single image. A notable characteristic of these SoCs is that the primary code is generally execute in place (XIP). Combined with the non-relocatable nature of the code, firmware updates need to be done in place.

7.2. Single CPU with Secure - Normal Mode Partitioning

Another configuration consists of a similar architecture to the previous, with a single CPU. However, this CPU supports a security partitioning scheme that allows memory (in addition to other things) to be divided into secure and normal mode. There will generally be two images, one for secure mode, and one for normal mode. In this configuration, firmware upgrades will generally be done by the CPU in secure mode, which is able to write to both areas of the flash device. In addition, there are requirements to be able to update either image independently, as well as to update them together atomically, as specified in the associated manifests.

7.3. Dual CPU, shared memory

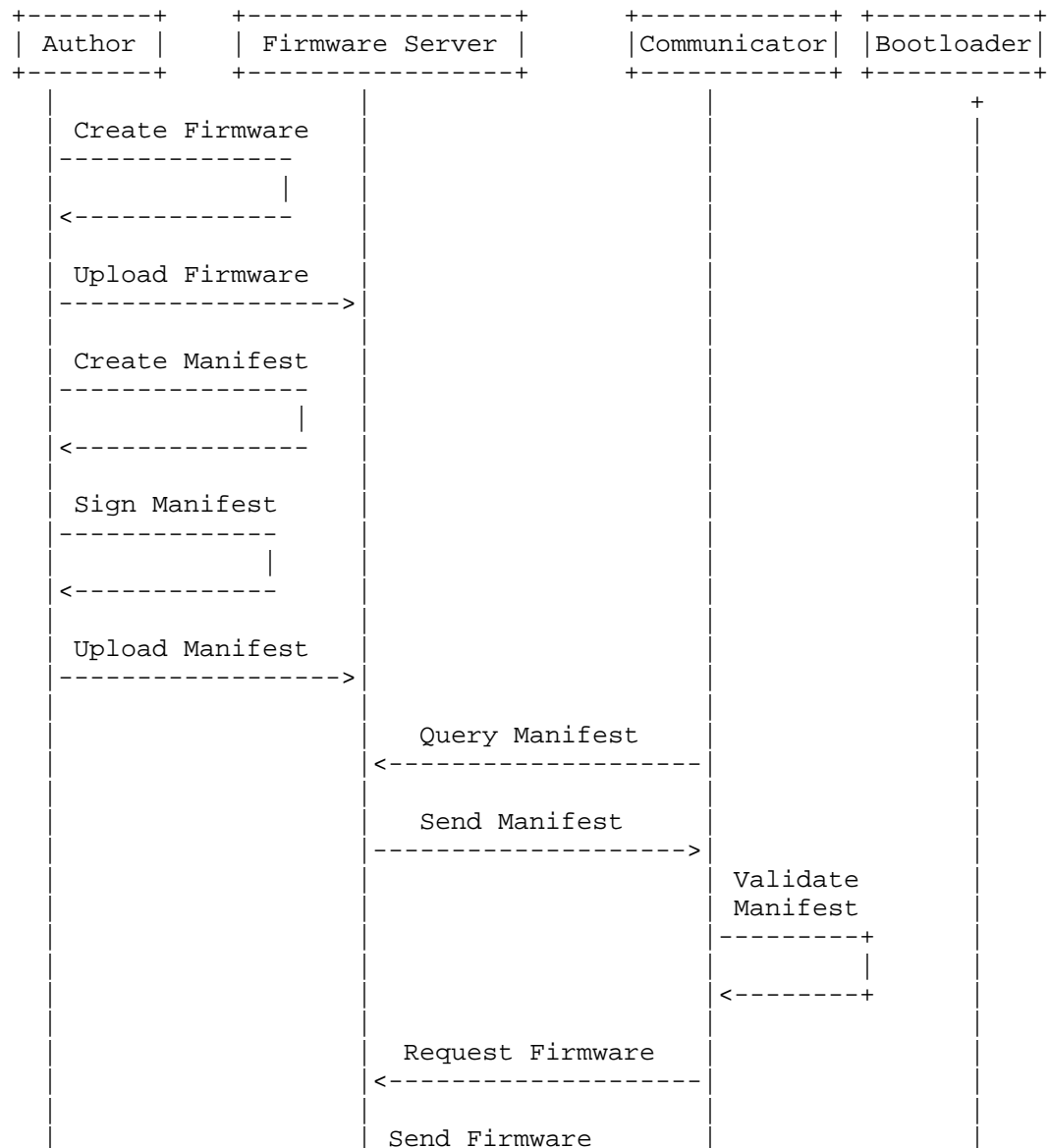
This configuration has two or more CPUs in a single SoC that share memory (flash and RAM). Generally, they will be a protection mechanism to prevent one CPU from accessing the other's memory. Upgrades in this case will typically be done by one of the CPUs, and is similar to the single CPU with secure mode.

7.4. Dual CPU, other bus

This configuration has two or more CPUs, each having their own memory. There will be a communication channel between them, but it will be used as a peripheral, not via shared memory. In this case, each CPU will have to be responsible for its own firmware upgrade. It is likely that one of the CPUs will be considered a master, and will direct the other CPU to do the upgrade. This configuration is commonly used to offload specific work to other CPUs. Firmware dependencies are similar to the other solutions above, sometimes allowing only one image to be upgraded, other times requiring several to be upgraded atomically. Because the updates are happening on multiple CPUs, upgrading the two images atomically is challenging.

8. Example Flow

The following example message flow illustrates the interaction for distributing a firmware image to a device starting with an author uploading the new firmware to Firmware Server and creating a manifest. The firmware and manifest are stored on the same Firmware Server.



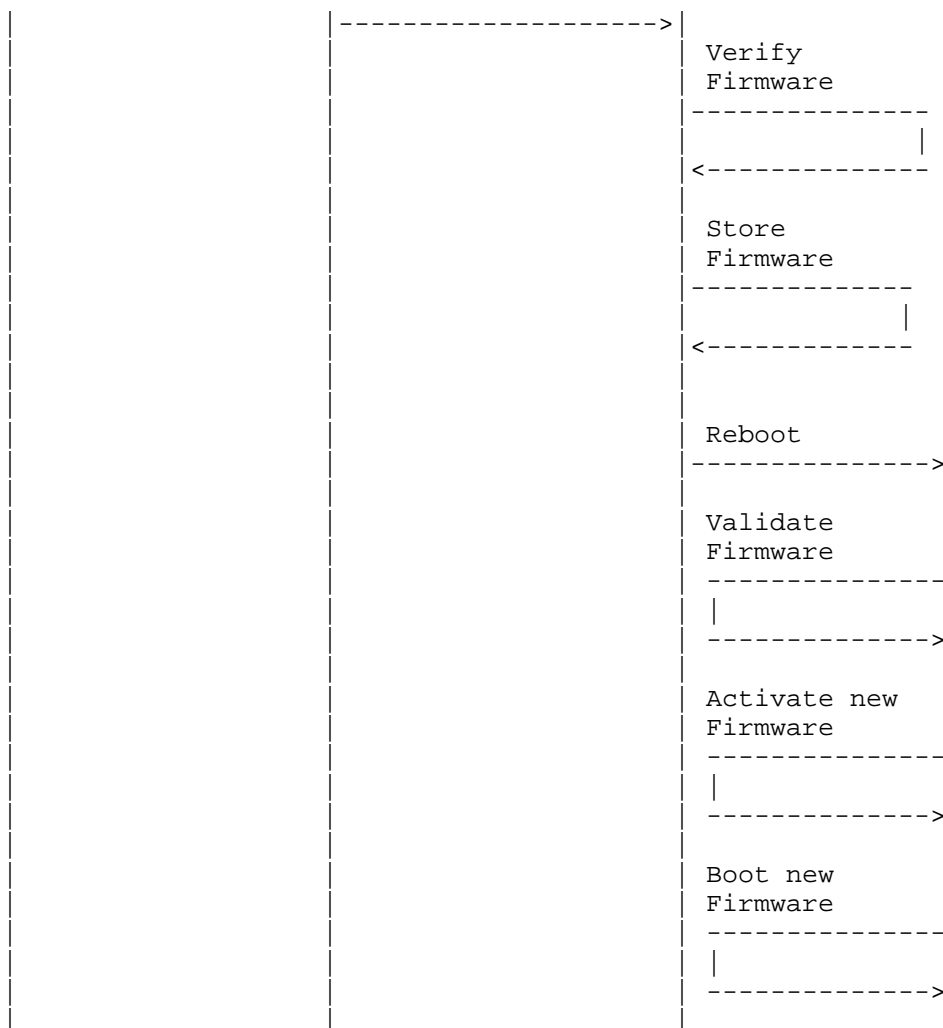


Figure 5: Example Flow for a Firmware Update.

9. IANA Considerations

This document does not require any actions by IANA.

10. Security Considerations

Firmware updates fix security vulnerabilities and are considered to be an important building block in securing IoT devices. Due to the importance of firmware updates for IoT devices the Internet Architecture Board (IAB) organized a 'Workshop on Internet of Things

(IoT) Software Update (IOTSU)', which took place at Trinity College Dublin, Ireland on the 13th and 14th of June, 2016 to take a look at the big picture. A report about this workshop can be found at [RFC8240]. A standardized firmware manifest format providing end-to-end security from the author to the device will be specified in a separate document.

There are, however, many other considerations raised during the workshop. Many of them are outside the scope of standardization organizations since they fall into the realm of product engineering, regulatory frameworks, and business models. The following considerations are outside the scope of this document, namely

- installing firmware updates in a robust fashion so that the update does not break the device functionality of the environment this device operates in.
- installing firmware updates in a timely fashion considering the complexity of the decision making process of updating devices, potential re-certification requirements, and the need for user consent to install updates.
- the distribution of the actual firmware update, potentially in an efficient manner to a large number of devices without human involvement.
- energy efficiency and battery lifetime considerations.
- key management required for verifying the digital signature protecting the manifest.
- incentives for manufacturers to offer a firmware update mechanism as part of their IoT products.

11. Mailing List Information

The discussion list for this document is located at the e-mail address suit@ietf.org [1]. Information on the group and information on how to subscribe to the list is at <https://www1.ietf.org/mailman/listinfo/suit>

Archives of the list can be found at: <https://www.ietf.org/mail-archive/web/suit/current/index.html>

12. Acknowledgements

We would like to thank the following persons for their feedback:

- Geraint Luff
- Amyas Phillips
- Dan Ros
- Thomas Eichinger
- Michael Richardson
- Emmanuel Baccelli
- Ned Smith
- Jim Schaad
- Carsten Bormann
- Cullen Jennings
- Olaf Bergmann
- Suhas Nandakumar
- Phillip Hallam-Baker
- Marti Bolivar
- Andrzej Puzdrowski
- Markus Gueller
- Henk Birkholz
- Jintao Zhu

We would also like to thank the WG chairs, Russ Housley, David Waltermire, Dave Thaler for their support and their reviews. Kathleen Moriarty was the responsible security area director when this work was started.

13. References

13.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC7925] Tschofenig, H., Ed. and T. Fossati, "Transport Layer Security (TLS) / Datagram Transport Layer Security (DTLS) Profiles for the Internet of Things", RFC 7925, DOI 10.17487/RFC7925, July 2016, <<https://www.rfc-editor.org/info/rfc7925>>.

13.2. Informative References

- [I-D.ietf-suit-information-model] Moran, B., Tschofenig, H., Birkholz, H., and J. Jimenez, "Firmware Updates for Internet of Things Devices - An Information Model for Manifests", draft-ietf-suit-information-model-00 (work in progress), June 2018.
- [LwM2M] OMA, ., "Lightweight Machine to Machine Technical Specification, Version 1.0.2", February 2018, <http://www.openmobilealliance.org/release/LightweightM2M/V1_0_2-20180209-A/OMA-TS-LightweightM2M-V1_0_2-20180209-A.pdf>.
- [RFC5649] Housley, R. and M. Dworkin, "Advanced Encryption Standard (AES) Key Wrap with Padding Algorithm", RFC 5649, DOI 10.17487/RFC5649, September 2009, <<https://www.rfc-editor.org/info/rfc5649>>.
- [RFC6024] Reddy, R. and C. Wallace, "Trust Anchor Management Requirements", RFC 6024, DOI 10.17487/RFC6024, October 2010, <<https://www.rfc-editor.org/info/rfc6024>>.
- [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>>.

13.3. URIs

- [1] <mailto:suit@ietf.org>

Authors' Addresses

Brendan Moran
Arm Limited

EMail: Brendan.Moran@arm.com

Milosch Meriac
Consultant

EMail: milosch@meriac.com

Hannes Tschofenig
Arm Limited

EMail: hannes.tschofenig@gmx.net

David Brown
Linaro

EMail: david.brown@linaro.org

SUIT
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2019

B. Moran
H. Tschofenig
Arm Limited
H. Birkholz
Fraunhofer SIT
July 02, 2018

Firmware Updates for Internet of Things Devices - An Information Model
for Manifests
draft-ietf-suit-information-model-01

Abstract

Vulnerabilities with Internet of Things (IoT) devices have raised the need for a solid and secure firmware update mechanism that is also suitable for constrained devices. Incorporating such update mechanism to fix vulnerabilities, to update configuration settings as well as adding new functionality is recommended by security experts.

One component of such a firmware update is the meta-data, or manifest, that describes the firmware image(s) and offers appropriate protection. This document describes all the information that must be present in the manifest.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 3, 2019.

Copyright Notice

Copyright (c) 2018 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. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	4
2. Conventions and Terminology	5
3. Motivation for Manifest Fields	5
3.1. Threat Model	5
3.2. Threat Descriptions	6
3.2.1. Threat MFT1: Old Firmware	6
3.2.2. Threat MFT2: Mismatched Firmware	6
3.2.3. Threat MFT3: Offline device + Old Firmware	7
3.2.4. Threat MFT4: The target device misinterprets the type of payload	7
3.2.5. Threat MFT5: The target device installs the payload to the wrong location	7
3.2.6. Threat MFT6: Redirection	8
3.2.7. Threat MFT7: Payload Verification on Boot	8
3.2.8. Threat MFT8: Unauthenticated Updates	8
3.2.9. Threat MFT9: Unexpected Precursor images	8
3.2.10. Threat MFT10: Unqualified Firmware	9
3.2.11. Threat MFT11: Reverse Engineering Of Firmware Image for Vulnerability Analysis	10
3.2.12. Threat MFT12: Overriding Critical Manifest Elements	10
3.3. Security Requirements	11
3.3.1. Security Requirement MFSR1: Monotonic Sequence Numbers	11
3.3.2. Security Requirement MFSR2: Vendor, Device-type	11

Identifiers	11
3.3.3. Security Requirement MFSR3: Best-Before Timestamps	11
3.3.4. Security Requirement MFSR5: Cryptographic Authenticity	12
3.3.5. Security Requirement MFSR4a: Authenticated Payload Type	12
3.3.6. Security Requirement MFSR4b: Authenticated Storage Location	12
3.3.7. Security Requirement MFSR4c: Authenticated Remote Resource Location	12
3.3.8. Security Requirement MFSR4d: Secure Boot	13
3.3.9. Security Requirement MFSR4e: Authenticated precursor images	13
3.3.10. Security Requirement MFSR4f: Authenticated Vendor and Class IDs	13
3.3.11. Security Requirement MFSR4f: Authenticated Vendor and Class IDs	13
3.3.12. Security Requirement MFSR6: Rights Require Authenticity	13
3.3.13. Security Requirement MFSR7: Firmware encryption	14
3.3.14. Security Requirement MFSR8: Access Control Lists	14
3.4. User Stories	14
3.4.1. Use Case MFUS1: Installation Instructions	15
3.4.2. Use Case MFUS2: Override Non-Critical Manifest Elements	15
3.4.3. Use Case MFUS3: Modular Update	16
3.4.4. Use Case MFUS4: Multiple Authorisations	16
3.4.5. Use Case MFUS5: Multiple Payload Formats	16
3.4.6. Use Case MFUS6: Prevent Confidential Information Disclosures	16
3.4.7. Use Case MFUS7: Prevent Devices from Unpacking Unknown Formats	16
3.4.8. Use Case MFUS8: Specify Version Numbers of Target Firmware	17
3.4.9. Use Case MFUS9: Enable devices to choose between images	17
3.5. Usability Requirements	17
3.5.1. Usability Requirement MFUR1	17
3.5.2. Usability Requirement MFUR2	17
3.5.3. Usability Requirement MFUR3	18
3.5.4. Usability Requirement MFUR4	19
3.5.5. Usability Requirement MFUR5	19
3.5.6. Usability Requirement MFUR6	19
3.5.7. Usability Requirement MFUR7	19
3.5.8. Usability Requirement MFUR8	20
4. Manifest Information Elements	20
4.1. Manifest Element: version identifier of the manifest structure	20

4.2.	Manifest Element: Monotonic Sequence Number	20
4.3.	Manifest Element: Vendor ID Condition	20
4.3.1.	Example: Domain Name-based UUIDs	21
4.4.	Manifest Element: Class ID Condition	21
4.4.1.	Example 1: Different Classes	21
4.4.2.	Example 2: Upgrading Class ID	22
4.4.3.	Example 3: Shared Functionality	22
4.5.	Manifest Element: Precursor Image Digest Condition	23
4.6.	Manifest Element: Required Image Version List	23
4.7.	Manifest Element: Best-Before timestamp condition	23
4.8.	Manifest Element: Payload Format	23
4.9.	Manifest Element: Processing Steps	24
4.10.	Manifest Element: Storage Location	24
4.10.1.	Example 1: Two Storage Locations	24
4.10.2.	Example 2: File System	24
4.10.3.	Example 3: Flash Memory	24
4.11.	Manifest Element: Component Identifier	25
4.12.	Manifest Element: URIs	25
4.13.	Manifest Element: Payload Digest	25
4.14.	Manifest Element: Size	25
4.15.	Manifest Element: Signature	26
4.16.	Manifest Element: Directives	26
4.17.	Manifest Element: Aliases	26
4.18.	Manifest Element: Dependencies	26
4.19.	Manifest Element: Content Key Distribution Method	27
4.20.	Manifest Element: XIP Address	27
5.	Security Considerations	27
6.	IANA Considerations	27
7.	Acknowledgements	27
8.	References	28
8.1.	Normative References	28
8.2.	Informative References	28
Appendix A.	Mailing List Information	29
Authors' Addresses	29

1. Introduction

The information model describes all the information elements required to secure firmware updates of IoT devices from the threats described in Section 3.1 and enable the user stories captured in Section 3.4. These threats and user stories are not intended to be an exhaustive list of the threats against IoT devices, nor of the possible use cases of firmware update; instead they are intended to describe the threats against firmware update in isolation and provide sufficient motivation to provide information elements that cover a wide range of use cases. The information model does not define the encoding, ordering, or structure of information elements, only their semantics.

Because the information model covers a wide range of user stories and a wide range of threats, not all information elements apply to all scenarios. As a result, many information elements could be considered optional to implement and optional to use, depending on which threats exist in a particular system and which use cases are required. Elements marked as mandatory provide baseline security and usability properties that are expected to be required for most applications. Those elements are mandatory to implement and mandatory to use. Elements marked as recommended provide important security or usability properties that are needed on most devices. Elements marked as optional enable security or usability properties that are useful in some applications.

2. Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

This document uses terms defined in [I-D.ietf-suit-architecture]. The term 'Operator' refers to both, Device and Network Operator.

3. Motivation for Manifest Fields

The following sub-sections describe the threat model, user stories, security requirements, and usability requirements.

3.1. Threat Model

The following sub-sections aim to provide information about the threats that were considered, the security requirements that are derived from those threats and the fields that permit implementation of the security requirements. This model uses the S.T.R.I.D.E. [STRIDE] approach. Each threat is classified according to:

- Spoofing Identity
- Tampering with data
- Repudiation
- Information disclosure
- Denial of service
- Elevation of privilege

This threat model only covers elements related to the transport of firmware updates. It explicitly does not cover threats outside of the transport of firmware updates. For example, threats to an IoT device due to physical access are out of scope.

3.2. Threat Descriptions

3.2.1. Threat MFT1: Old Firmware

Classification: Elevation of Privilege

An attacker sends an old, but valid manifest with an old, but valid firmware image to a device. If there is a known vulnerability in the provided firmware image, this may allow an attacker to exploit the vulnerability and gain control of the device.

Threat Escalation: If the attacker is able to exploit the known vulnerability, then this threat can be escalated to ALL TYPES.

Mitigated by: MFSR1

3.2.2. Threat MFT2: Mismatched Firmware

Classification: Denial of Service

An attacker sends a valid firmware image, for the wrong type of device, signed by an actor with firmware installation permission on both types of device. The firmware is verified by the device positively because it is signed by an actor with the appropriate permission. This could have wide-ranging consequences. For devices that are similar, it could cause minor breakage, or expose security vulnerabilities. For devices that are very different, it is likely to render devices inoperable.

Mitigated by: MFSR2

Example:

Suppose that two vendors, Vendor A and Vendor B, adopt the same trade name in different geographic regions, and they both make products with the same names, or product name matching is not used. This causes firmware from Vendor A to match devices from Vendor B.

If the vendors are the firmware authorities, then devices from Vendor A will reject images signed by Vendor B since they use different credentials. However, if both devices trust the same firmware authority, then, devices from Vendor A could install firmware intended for devices from Vendor B.

3.2.3. Threat MFT3: Offline device + Old Firmware

Classification: Elevation of Privilege

An attacker targets a device that has been offline for a long time and runs an old firmware version. The attacker sends an old, but valid manifest to a device with an old, but valid firmware image. The attacker-provided firmware is newer than the installed one but older than the most recently available firmware. If there is a known vulnerability in the provided firmware image then this may allow an attacker to gain control of a device. Because the device has been offline for a long time, it is unaware of any new updates. As such it will treat the old manifest as the most current.

Threat Escalation: If the attacker is able to exploit the known vulnerability, then this threat can be escalated to ALL TYPES.

Mitigated by: MFSR3

3.2.4. Threat MFT4: The target device misinterprets the type of payload

Classification: Denial of Service

If a device misinterprets the type of the firmware image, it may cause a device to install a firmware image incorrectly. An incorrectly installed firmware image would likely cause the device to stop functioning.

Threat Escalation: An attacker that can cause a device to misinterpret the received firmware image may gain elevation of privilege and potentially expand this to all types of threat.

Mitigated by: MFSR4a

3.2.5. Threat MFT5: The target device installs the payload to the wrong location

Classification: Denial of Service

If a device installs a firmware image to the wrong location on the device, then it is likely to break. For example, a firmware image installed as an application could cause a device and/or an application to stop functioning.

Threat Escalation: An attacker that can cause a device to misinterpret the received code may gain elevation of privilege and potentially expand this to all types of threat.

Mitigated by: MFSR4b

3.2.6. Threat MFT6: Redirection

Classification: Denial of Service

If a device does not know where to obtain the payload for an update, it may be redirected to an attacker's server. This would allow an attacker to provide broken payloads to devices.

Mitigated by: MFSR4c

3.2.7. Threat MFT7: Payload Verification on Boot

Classification: Elevation of Privilege

An attacker replaces a newly downloaded firmware after a device finishes verifying a manifest. This could cause the device to execute the attacker's code. This attack likely requires physical access to the device. However, it is possible that this attack is carried out in combination with another threat that allows remote execution.

Threat Escalation: If the attacker is able to exploit a known vulnerability, or if the attacker can supply their own firmware, then this threat can be escalated to ALL TYPES.

Mitigated by: MFSR4d

3.2.8. Threat MFT8: Unauthenticated Updates

Classification: Elevation of Privilege

If an attacker can install their firmware on a device, by manipulating either payload or metadata, then they have complete control of the device.

Threat Escalation: If the attacker is able to exploit a known vulnerability, or if the attacker can supply their own firmware, then this threat can be escalated to ALL TYPES.

Mitigated by: MFSR5

3.2.9. Threat MFT9: Unexpected Precursor images

Classification: Denial of Service

An attacker sends a valid, current manifest to a device that has an unexpected precursor image. If a payload format requires a precursor image (for example, delta updates) and that precursor image is not available on the target device, it could cause the update to break.

Threat Escalation: An attacker that can cause a device to install a payload against the wrong precursor image could gain elevation of privilege and potentially expand this to all types of threat.

Mitigated by: MFSR4e

3.2.10. Threat MFT10: Unqualified Firmware

Classification: Denial of Service, Elevation of Privilege

This threat can appear in several ways, however it is ultimately about interoperability of devices with other systems. The owner or operator of a network needs to approve firmware for their network in order to ensure interoperability with other devices on the network, or the network itself. If the firmware is not qualified, it may not work. Therefore, if a device installs firmware without the approval of the network owner or operator, this is a threat to devices and the network.

Threat Escalation: If the firmware expects configuration that is present in devices deployed in Network A, but not in devices deployed in Network B, then the device may experience degraded security, leading to threats of All Types.

Mitigated by: MFSR6, MFSR8

3.2.10.1. Example 1: Multiple Network Operators with a Single Device Operator

In this example let us assume that Device Operators expect the rights to create firmware but that Network Operators expect the rights to qualify firmware as fit-for-purpose on their networks. Additionally assume that an Device Operators manage devices that can be deployed on any network, including Network A and B in our example.

An attacker may obtain a manifest for a device on Network A. Then, this attacker sends that manifest to a device on Network B. Because Network A and Network B are under control of different Operators, and the firmware for a device on Network A has not been qualified to be deployed on Network B, the target device on Network B is now in violation of the Operator B's policy and may get disabled by this unqualified, but signed firmware.

This is a denial of service because it can render devices inoperable. This is an elevation of privilege because it allows the attacker to make installation decisions that should be made by the Operator.

3.2.10.2. Example 2: Single Network Operator with Multiple Device Operators

Multiple devices that interoperate are used on the same network and communicate with each other. Some devices are manufactured and managed by Device Operator A and other devices by Device Operator B. A new firmware is released by Device Operator A that breaks compatibility with devices from Device Operator B. An attacker sends the new firmware to the devices managed by Device Operator A without approval of the Network Operator. This breaks the behaviour of the larger system causing denial of service and possibly other threats. Where the network is a distributed SCADA system, this could cause misbehaviour of the process that is under control.

3.2.11. Threat MFT11: Reverse Engineering Of Firmware Image for Vulnerability Analysis

Classification: All Types

An attacker wants to mount an attack on an IoT device. To prepare the attack he or she retrieves the provided firmware image and performs reverse engineering of the firmware image to analyze it for specific vulnerabilities.

Mitigated by: MFSR7

3.2.12. Threat MFT12: Overriding Critical Manifest Elements

Classification: Elevation of Privilege

An authorised actor, but not the firmware authority, uses an override mechanism (MFUS2) to change an information element in a manifest signed by the firmware authority. For example, if the authorised actor overrides the digest and URI of the payload, the actor can replace the entire payload with a payload of their choice.

Threat Escalation: By overriding elements such as payload installation instructions or firmware digest, this threat can be escalated to all types.

Mitigated by: MFSR8

3.3. Security Requirements

The security requirements here are a set of policies that mitigate the threats described in Section 3.1.

3.3.1. Security Requirement MFSR1: Monotonic Sequence Numbers

Only an actor with firmware installation authority is permitted to decide when device firmware can be installed. To enforce this rule, manifests **MUST** contain monotonically increasing sequence numbers. Manifests **MAY** use UTC epoch timestamps to coordinate monotonically increasing sequence numbers across many actors in many locations. If UTC epoch timestamps are used, they **MUST NOT** be treated as times, they **MUST** be treated only as sequence numbers. Devices **MUST** reject manifests with sequence numbers smaller than any onboard sequence number.

Note: This is not a firmware version. It is a manifest sequence number. A firmware version may be rolled back by creating a new manifest for the old firmware version with a later sequence number.

Mitigates: Threat MFT1

Implemented by: Manifest Element: Monotonic Sequence Number

3.3.2. Security Requirement MFSR2: Vendor, Device-type Identifiers

Devices **MUST** only apply firmware that is intended for them. Devices **MUST** know with fine granularity that a given update applies to their vendor, model, hardware revision, software revision. Human-readable identifiers are often error-prone in this regard, so unique identifiers **SHOULD** be used.

Mitigates: Threat MFT2

Implemented by: Manifest Elements: Vendor ID Condition, Class ID Condition

3.3.3. Security Requirement MFSR3: Best-Before Timestamps

Firmware **MAY** expire after a given time. Devices **MAY** provide a secure clock (local or remote). If a secure clock is provided and the Firmware manifest has a best-before timestamp, the device **MUST** reject the manifest if current time is larger than the best-before time.

Mitigates: Threat MFT3

Implemented by: Manifest Element: Best-Before timestamp condition

3.3.4. Security Requirement MFSR5: Cryptographic Authenticity

The authenticity of an update must be demonstrable. Typically, this means that updates must be digitally authenticated. Because the manifest contains information about how to install the update, the manifest's authenticity must also be demonstrable. To reduce the overhead required for validation, the manifest contains the digest of the firmware image, rather than a second digital signature. The authenticity of the manifest can be verified with a digital signature or Message Authentication Code, the authenticity of the firmware image is tied to the manifest by the use of a digest of the firmware image.

Mitigates: Threat MFT8

Implemented by: Signature, Payload Digest

3.3.5. Security Requirement MFSR4a: Authenticated Payload Type

The type of payload (which may be independent of format) MUST be authenticated. For example, the target must know whether the payload is XIP firmware, a loadable module, or serialized configuration data.

Mitigates: MFT4

Implemented by: Manifest Elements: Payload Format, Storage Location

3.3.6. Security Requirement MFSR4b: Authenticated Storage Location

The location on the target where the payload is to be stored MUST be authenticated.

Mitigates: MFT5

Implemented by: Manifest Elements: Storage Location

3.3.7. Security Requirement MFSR4c: Authenticated Remote Resource Location

The location where a target should find a payload MUST be authenticated.

Mitigates: MFT6

Implemented by: Manifest Elements: URIs

3.3.8. Security Requirement MFSR4d: Secure Boot

The target SHOULD verify firmware at time of boot. This requires authenticated payload size, and digest.

Mitigates: MFT7

Implemented by: Manifest Elements: Payload Digest, Size

3.3.9. Security Requirement MFSR4e: Authenticated precursor images

If an update uses a differential compression method, it MUST specify the digest of the precursor image and that digest MUST be authenticated.

Mitigates: MFT9

Implemented by: Manifest Elements: Precursor Image Digest Condition

3.3.10. Security Requirement MFSR4f: Authenticated Vendor and Class IDs

The identifiers that specify firmware compatibility MUST be authenticated to ensure that only compatible firmware is installed on a target device.

Mitigates: MFT2

Implemented By: Manifest Elements: Vendor ID Condition, Class ID Condition

3.3.11. Security Requirement MFSR4f: Authenticated Vendor and Class IDs

The identifiers that specify firmware compatibility MUST be authenticated to ensure that only compatible firmware is installed on a target device.

Mitigates: MFT2

Implemented By: Manifest Elements: Vendor ID Condition, Class ID Condition

3.3.12. Security Requirement MFSR6: Rights Require Authenticity

If a device grants different rights to different actors, exercising those rights MUST be accompanied by proof of those rights, in the form of proof of authenticity. Authenticity mechanisms such as those required in MFSR5 are acceptable but need to follow the end-to-end security model.

For example, if a device has a policy that requires that firmware have both an Authorship right and a Qualification right and if that device grants Authorship and Qualification rights to different parties, such as a Device Operator and a Network Operator, respectively, then the firmware cannot be installed without proof of rights from both the Device and the Network Operator.

Mitigates: MFT10

Implemented by: Signature

3.3.13. Security Requirement MFSR7: Firmware encryption

The manifest information model must enable encrypted payloads. Encryption helps to prevent third parties, including attackers, from reading the content of the firmware image. This can protect against confidential information disclosures and discovery of vulnerabilities through reverse engineering. Therefore the manifest must convey the information required to allow an intended recipient to decrypt an encrypted payload.

Mitigates: MFT11

Implemented by: Manifest Element: Content Key Distribution Method

3.3.14. Security Requirement MFSR8: Access Control Lists

If a device grants different rights to different actors, then an exercise of those rights must be validated against a list of rights for the actor. This typically takes the form of an Access Control List (ACL). ACLs are applied to two scenarios:

1. An ACL decides which elements of the manifest may be overridden and by which actors.
2. An ACL decides which component identifier/storage identifier pairs can be written by which actors.

Mitigates: MFT12, MFT10

Implemented by: Client-side code, not specified in manifest.

3.4. User Stories

User stories provide expected use cases. These are used to feed into usability requirements.

3.4.1. Use Case MFUS1: Installation Instructions

As an Device Operator, I want to provide my devices with additional installation instructions so that I can keep process details out of my payload data.

Some installation instructions might be:

- Use a table of hashes to ensure that each block of the payload is validate before writing.
- Do not report progress.
- Pre-cache the update, but do not install.
- Install the pre-cached update matching this manifest.
- Install this update immediately, overriding any long-running tasks.

Satisfied by: MFUR1

3.4.2. Use Case MFUS2: Override Non-Critical Manifest Elements

As a Network Operator, I would like to be able to override the non-critical information in the manifest so that I can control my devices more precisely. This assumes that the Device Operator delegated rights about the device to the Network Operator.

Some examples of potentially overridable information:

- URIs: this allows the Network Operator to direct devices to their own infrastructure in order to reduce network load.
- Conditions: this allows the Network Operator to pose additional constraints on the installation of the manifest.
- Directives: this allows the Network Operator to add more instructions such as time of installation.
- Processing Steps: If an intermediary performs an action on behalf of a device, it may need to override the processing steps. It is still possible for a device to verify the final content and the result of any processing step that specifies a digest. Some processing steps should be non-overridable.

Satisfied by: MFUR2, MFUR3

3.4.3. Use Case MFUS3: Modular Update

As an Operator, I want to divide my firmware into frequently updated and infrequently updated components, so that I can reduce the size of updates and make different parties responsible for different components.

Satisfied by: MFUR3

3.4.4. Use Case MFUS4: Multiple Authorisations

As a Device Operator, I want to ensure the quality of a firmware update before installing it, so that I can ensure interoperability of all devices in my product family. I want to restrict the ability to make changes to my devices to require my express approval.

Satisfied by: MFUR4, MFSR8

3.4.5. Use Case MFUS5: Multiple Payload Formats

As an Operator, I want to be able to send multiple payload formats to suit the needs of my update, so that I can optimise the bandwidth used by my devices.

Satisfied by: MFUR5

3.4.6. Use Case MFUS6: Prevent Confidential Information Disclosures

As an firmware author, I want to prevent confidential information from being disclosed during firmware updates. It is assumed that channel security is adequate to protect the manifest itself against information disclosure.

Satisfied by: MFSR7

3.4.7. Use Case MFUS7: Prevent Devices from Unpacking Unknown Formats

As a Device Operator, I want devices to determine whether they can process a payload prior to downloading it.

In some cases, it may be desirable for a third party to perform some processing on behalf of a target. For this to occur, the third party MUST indicate what processing occurred and how to verify it against the Trust Provisioning Authority's intent.

This amounts to overriding Processing Steps and URIs.

Satisfied by: MFUR6, MFUR2

3.4.8. Use Case MFUS8: Specify Version Numbers of Target Firmware

As a Device Operator, I want to be able to target devices for updates based on their current firmware version, so that I can control which versions are replaced with a single manifest.

Satisfied by: MFUR7

3.4.9. Use Case MFUS9: Enable devices to choose between images

As a developer, I want to be able to sign two or more versions of my firmware in a single manifest so that I can use a very simple bootloader that chooses between two or more images that are executed in-place.

Satisfied by: MFUR8

3.5. Usability Requirements

The following usability requirements satisfy the user stories listed above.

3.5.1. Usability Requirement MFUR1

It must be possible to provide all information necessary for the processing of a manifest into the manifest.

Satisfies: User story MFUS1

Implemented by: Manifest Element: Directives

3.5.2. Usability Requirement MFUR2

It must be possible to redirect payload fetches. This applies where two manifests are used in conjunction. For example, a Device Operator creates a manifest specifying a payload and signs it, and provides a URI for that payload. A Network Operator creates a second manifest, with a dependency on the first. They use this second manifest to override the URIs provided by the Device Operator, directing them into their own infrastructure instead. Some devices may provide this capability, while others may only look at canonical sources of firmware. For this to be possible, the device must fetch the payload, whereas a device that accepts payload pushes will ignore this feature.

Satisfies: User story MFUS2

Implemented by: Manifest Element: Aliases

3.5.3. Usability Requirement MFUR3

It must be possible express the requirement to install one or more payloads from one or more authorities so that a multi-payload update can be described. This allows multiple parties with different permissions to collaborate in creating a single update for the IoT device, across multiple components.

This requirement effectively means that it must be possible to construct a tree of manifests on a multi-image target.

Because devices can be either HeSA or HoSA both the storage system and the storage location within that storage system must be possible to specify. In a HoSA device, the payload location may be as simple as an address, or a file path. In a HeSA device, the payload location may be scoped by a component identifier. It is expedient to consider that all HoSA devices are HeSA devices with a single component.

3.5.3.1. Example 1: Multiple Microcontrollers

An IoT device with multiple microcontrollers in the same physical device (HeSA) will likely require multiple payloads with different component identifiers.

3.5.3.2. Example 2: Code and Configuration

A firmware image can be divided into two payloads: code and configuration. These payloads may require authorizations from different actors in order to install (see MFSR6 and MFSR8). This structure means that multiple manifests may be required, with a dependency structure between them.

3.5.3.3. Example 3: Multiple Chunks

A firmware image can be divided into multiple functional blocks for separate testing and distribution. This means that code would need to be distributed in multiple payloads. For example, this might be desirable in order to ensure that common code between devices is identical in order to reduce distribution bandwidth.

Satisfies: User story MFUS2, MFUS3

Implemented by Manifest Element: Dependencies, StorageIdentifier, ComponentIdentifier

3.5.4. Usability Requirement MFUR4

It MUST be possible to sign a manifest multiple times so that signatures from multiple parties with different permissions can be required in order to authorise installation of a manifest.

Satisfies: User story MFUS4

Implemented by: COSE Signature (or similar)

3.5.5. Usability Requirement MFUR5

The manifest format MUST accommodate any payload format that an Operator wishes to use. Some examples of payload format would be:

- Binary
- Elf
- Differential
- Compressed
- Packed configuration
- Intel HEX
- S-Record

Satisfies: User story MFUS5

Implemented by: Manifest Element: Payload Format

3.5.6. Usability Requirement MFUR6

The manifest format must accommodate nested formats, announcing to the target device all the nesting steps and any parameters used by those steps.

Satisfies: User story MFUS6

Implemented by: Manifest Element: Processing Steps

3.5.7. Usability Requirement MFUR7

The manifest format must provide a method to specify multiple version numbers of firmware to which the manifest applies, either with a list or with range matching.

Satisfies: User story MFUS8

Implemented by: Manifest Element: Required Image Version List

3.5.8. Usability Requirement MFUR8

The manifest format must provide a mechanism to list multiple equivalent payloads by Execute-In-Place Installation Address, including the payload digest and, optionally, payload URIs.

Satisfies: User story MFUS9

Implemented by: Manifest Element: XIP Address

4. Manifest Information Elements

Each manifest element is anchored in a security requirement or a usability requirement. The manifest elements are described below and justified by their requirements.

4.1. Manifest Element: version identifier of the manifest structure

An identifier that describes which iteration of the manifest format is contained in the structure.

This element is MANDATORY and must be present in order to allow devices to identify the version of the manifest data model that is in use.

4.2. Manifest Element: Monotonic Sequence Number

A monotonically increasing sequence number. For convenience, the monotonic sequence number MAY be a UTC timestamp. This allows global synchronisation of sequence numbers without any additional management.

This element is MANDATORY and is necessary to prevent malicious actors from reverting a firmware update against the wishes of the relevant authority.

Implements: Security Requirement MFSR1.

4.3. Manifest Element: Vendor ID Condition

Vendor IDs MUST be unique. This is to prevent similarly, or identically named entities from different geographic regions from colliding in their customer's infrastructure. Recommended practice

is to use type 5 UUIDs with the vendor's domain name and the UUID DNS prefix. Other options include type 1 and type 4 UUIDs.

This ID is OPTIONAL but RECOMMENDED and helps to distinguish between identically named products from different vendors.

Implements: Security Requirement MFSR2, MFSR4f.

4.3.1. Example: Domain Name-based UUIDs

Vendor A creates a UUID based on their domain name:

```
vendorId = UUID5(DNS, "vendor-a.com")
```

Because the DNS infrastructure prevents multiple registrations of the same domain name, this UUID is guaranteed to be unique. Because the domain name is known, this UUID is reproducible. Type 1 and type 4 UUIDs produce similar guarantees of uniqueness, but not reproducibility.

4.4. Manifest Element: Class ID Condition

A device "Class" is defined as any device that can accept the same firmware update without modification. Class Identifiers MUST be unique within a Vendor ID. This is to prevent similarly, or identically named devices colliding in their customer's infrastructure. Recommended practice is to use type 5 UUIDs with the model, hardware revision, etc. and use the Vendor ID as the UUID prefix. Other options include type 1 and type 4 UUIDs. Classes MAY be implemented in a more granular way. Classes MUST NOT be implemented in a less granular way. Class ID can encompass model name, hardware revision, software revision. Devices MAY have multiple Class IDs.

Note Well: Class ID is not a human-readable element. It is intended for match/mismatch use only.

This ID is OPTIONAL but RECOMMENDED and allows devices to determine applicability of a firmware in an unambiguous way.

Implements: Security Requirement MFSR2, MFSR4f.

4.4.1. Example 1: Different Classes

Vendor A creates product Z and product Y. The firmware images of products Z and Y are not interchangeable. Vendor A creates UUIDs as follows:

- vendorId = UUID5(DNS, "vendor-a.com")
- ZclassId = UUID5(vendorId, "Product Z")
- YclassId = UUID5(vendorId, "Product Y")

This ensures that Vendor A's Product Z cannot install firmware for Product Y and Product Y cannot install firmware for Product Z.

4.4.2. Example 2: Upgrading Class ID

Vendor A creates product X. Later, Vendor A adds a new feature to product X, creating product X v2. Product X requires a firmware update to work with firmware intended for product X v2.

Vendor A creates UUIDs as follows:

- vendorId = UUID5(DNS, "vendor-a.com")
- XclassId = UUID5(vendorId, "Product X")
- Xv2classId = UUID5(vendorId, "Product X v2")

When product X receives the firmware update necessary to be compatible with product X v2, part of the firmware update changes the class ID to Xv2classId.

4.4.3. Example 3: Shared Functionality

Vendor A produces two products, product X and product Y. These components share a common core (such as an operating system), but have different applications. The common core and the applications can be updated independently. To enable X and Y to receive the same common core update, they require the same class ID. To ensure that only product X receives application X and only product Y receives application Y, product X and product Y must have different class IDs. The vendor creates Class IDs as follows:

- vendorId = UUID5(DNS, "vendor-a.com")
- XclassId = UUID5(vendorId, "Product X")
- YclassId = UUID5(vendorId, "Product Y")
- CommonClassId = UUID5(vendorId, "common core")

Product X matches against both XclassId and CommonClassId. Product Y matches against both YclassId and CommonClassId.

4.5. Manifest Element: Precursor Image Digest Condition

When a precursor image is required by the payload format, a precursor image digest condition MUST be present in the conditions list. The precursor image may be installed or stored as a candidate.

This element is MANDATORY for differential updates. Otherwise, it is not needed.

Implements: Security Requirement MFSR4e

4.6. Manifest Element: Required Image Version List

When a payload applies to multiple versions of a firmware, the required image version list specifies which versions must be present for the update to be applied. This allows the update author to target specific versions of firmware for an update, while excluding those to which it should not be applied.

Where an update can only be applied over specific predecessor versions, that version MUST be specified by the Required Image Version List.

This element is OPTIONAL.

Implements: MFUR7

4.7. Manifest Element: Best-Before timestamp condition

This element tells a device the last application time. This is only usable in conjunction with a secure clock.

This element is OPTIONAL and MAY enable use cases where a secure clock is provided and firmware is intended to expire regularly.

Implements: Security Requirement MFSR3

4.8. Manifest Element: Payload Format

The format of the payload must be indicated to devices in an unambiguous way. This element provides a mechanism to describe the payload format, within the signed metadata.

This element is MANDATORY and MUST be present to enable devices to decode payloads correctly.

Implements: Security Requirement MFSR4a, Usability Requirement MFUR5

4.9. Manifest Element: Processing Steps

A list of all payload processors necessary to process a nested format and any parameters needed by those payload processors. Each Processing Step SHOULD indicate the expected digest of the payload after the processing is complete. Processing steps are distinct from Directives in that Directives apply to the manifest as a whole, whereas Processing Steps apply to an individual payload and provide instructions on how to unpack it.

Implements: Usability Requirement MFUR6

4.10. Manifest Element: Storage Location

This element tells the device which component is being updated. The device can use this to establish which permissions are necessary and the physical location to use.

This element is MANDATORY and MUST be present to enable devices to store payloads to the correct location.

Implements: Security Requirement MFSR4b

4.10.1. Example 1: Two Storage Locations

A device supports two components: an OS and an application. These components can be updated independently, expressing dependencies to ensure compatibility between the components. The firmware authority chooses two storage identifiers:

- OS
- APP

4.10.2. Example 2: File System

A device supports a full filesystem. The firmware authority chooses to make the storage identifier the path at which to install the payload. The payload may be a tarball, in which case, it unpacks the tarball into the specified path.

4.10.3. Example 3: Flash Memory

A device supports flash memory. The firmware authority chooses to make the storage identifier the offset where the image should be written.

4.11. Manifest Element: Component Identifier

In a heterogeneous storage architecture, a storage identifier is insufficient to identify where and how to store a payload. To resolve this, a component identifier indicates which part of the storage architecture is targeted by the payload. In a homogeneous storage architecture, this element is unnecessary.

This element is OPTIONAL and only necessary in heterogeneous storage architecture devices.

Implements: MFUR3

4.12. Manifest Element: URIs

This element is a list of weighted URIs that the device uses to select where to obtain a payload.

This element is OPTIONAL and only needed when the target device does not intrinsically know where to find the payload.

Note: Devices will typically require URIs.

Implements: Security Requirement MFSR4c

4.13. Manifest Element: Payload Digest

This element contains the digest of the payload. This allows the target device to ensure authenticity of the payload. It MUST be possible to specify more than one payload digest, indexed by Manifest Element: XIP Address.

This element is MANDATORY and fundamentally necessary to ensure the authenticity and integrity of the payload.

Implements: Security Requirement MFSR4d, Usability Requirement MFUR8

4.14. Manifest Element: Size

The size of the payload in bytes.

This element is MANDATORY and informs the target device how big of a payload to expect. Without it, devices are exposed to some classes of denial of service attack.

Implements: Security Requirement MFSR4d

4.15. Manifest Element: Signature

This is not strictly a manifest element. Instead, the manifest is wrapped by a standardised authentication container, such as a COSE or CMS signature object. The authentication container **MUST** support multiple actors and multiple authentications.

This element is **MANDATORY** and represents the foundation of all security properties of the manifest.

Implements: Security Requirement MFSR5, MFSR6, MFUR4

4.16. Manifest Element: Directives

A list of instructions that the device should execute, in order, when processing the manifest. This information is distinct from the information necessary to process a payload (Processing Steps) and applies to the whole manifest including all payloads that it references. Directives include information such as update timing (For example, install only on Sunday, at 0200), procedural considerations (for example, shut down the equipment under control before executing the update), pre and post-installation steps (for example, run a script).

This element is **OPTIONAL** and enables some use cases.

Implements: Usability Requirement MFUR1

4.17. Manifest Element: Aliases

A list of Digest/URI pairs. A device should build an alias table while parsing a manifest tree and treat any aliases as top-ranked URIs for the corresponding digest.

This element is **OPTIONAL** and enables some use cases.

Implements: Usability Requirement MFUR2

4.18. Manifest Element: Dependencies

A list of Digest/URI pairs that refer to other manifests by digest. The manifests that are linked in this way must be acquired and installed simultaneously in order to form a complete update.

This element is **MANDATORY** to use in deployments that include both multiple authorities and multiple payloads.

Implements: Usability Requirement MFUR3

4.19. Manifest Element: Content Key Distribution Method

Encrypting firmware images requires symmetric content encryption keys. Since there are several methods to protect or distribute the symmetric content encryption keys, the manifest contains a element for the Content Key Distribution Method. One examples for such a Content Key Distribution Method is the usage of Key Tables, pointing to content encryption keys, which themselves are encrypted using the public keys of devices. This MAY be included in a decryption step contained in Processing Steps.

This element is MANDATORY to use for encrypted payloads,

Implements: Security Requirement MFSR7.

4.20. Manifest Element: XIP Address

In order to support XIP systems with multiple possible base addresses, it is necessary to specify which address the payload is linked for.

For example a microcontroller may have a simple bootloader that chooses one of two images to boot. That microcontroller then needs to choose one of two firmware images to install, based on which of its two images is older.

Implements: MFUR8

5. Security Considerations

Security considerations for this document are covered in Section 3.

6. IANA Considerations

This document does not require any actions by IANA.

7. Acknowledgements

We would like to thank our working group chairs, Dave Thaler, Russ Housley and David Waltermire, for their review comments and their support.

We would like to thank the participants of the 2018 Berlin SUIT Hackathon and the June 2018 virtual design team meetings for their discussion input. In particular, we would like to thank Koen Zandberg, Emmanuel Baccelli, Carsten Bormann, David Brown, Markus Gueller, Frank Audun Kvamtro, Oyvind Ronningstad, Michael Richardson, Jan-Frederik Rieckers Francisco Acosta, Anton Gerasimov, Matthias

Waehtlich, Max Groening, Daniel Petry, Gaetan Harter, Ralph Hamm, Steve Patrick, Fabio Utzig, Paul Lambert, Benjamin Kaduk, Said Gharout, and Milen Stoychev.

8. References

8.1. Normative References

- [I-D.ietf-suit-architecture]
Moran, B., Meriac, M., Tschofenig, H., and D. Brown, "A Firmware Update Architecture for Internet of Things Devices", draft-ietf-suit-architecture-01 (work in progress), July 2018.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

8.2. Informative References

- [RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, DOI 10.17487/RFC4122, July 2005, <<https://www.rfc-editor.org/info/rfc4122>>.
- [STRIDE] Microsoft, "The STRIDE Threat Model", May 2018, <[https://msdn.microsoft.com/en-us/library/ee823878\(v=cs.20\).aspx](https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx)>.

8.3. URIs

- [1] <mailto:suit@ietf.org>

Appendix A. Mailing List Information

The discussion list for this document is located at the e-mail address suit@ietf.org [1]. Information on the group and information on how to subscribe to the list is at <https://www1.ietf.org/mailman/listinfo/suit>

Archives of the list can be found at: <https://www.ietf.org/mail-archive/web/suit/current/index.html>

Authors' Addresses

Brendan Moran
Arm Limited

EMail: Brendan.Moran@arm.com

Hannes Tschofenig
Arm Limited

EMail: hannes.tschofenig@gmx.net

Henk Birkholz
Fraunhofer SIT

EMail: henk.birkholz@sit.fraunhofer.de

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: February 3, 2020

J. Fenton
Altmode Networks
August 2, 2019

SMTP Require TLS Option
draft-ietf-uta-smtp-require-tls-09

Abstract

The SMTP STARTTLS option, used in negotiating transport-level encryption of SMTP connections, is not as useful from a security standpoint as it might be because of its opportunistic nature; message delivery is, by default, prioritized over security. This document describes an SMTP service extension, REQUIRETLS, and message header field, TLS-Required. If the REQUIRETLS option or TLS-Required message header field is used when sending a message, it asserts a request on the part of the message sender to override the default negotiation of TLS, either by requiring that TLS be negotiated when the message is relayed, or by requesting that recipient-side policy mechanisms such as MTA-STS and DANE be ignored when relaying a message for which security is unimportant.

Status of This Memo

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

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

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

This Internet-Draft will expire on February 3, 2020.

Copyright Notice

Copyright (c) 2019 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

(<https://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. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Requirements Language	4
2. The REQUIRETLS Service Extension	4
3. The TLS-Required Header Field	5
4. REQUIRETLS Semantics	6
4.1. REQUIRETLS Receipt Requirements	6
4.2. REQUIRETLS Sender Requirements	6
4.2.1. Sending with TLS Required	6
4.2.2. Sending with TLS Optional	7
4.3. REQUIRETLS Submission	8
4.4. Delivery of REQUIRETLS messages	8
5. Non-delivery message handling	8
6. Reorigination considerations	9
7. IANA Considerations	10
8. Security Considerations	11
8.1. Passive attacks	11
8.2. Active attacks	11
8.3. Bad Actor MTAs	12
8.4. Policy Conflicts	12
9. Acknowledgements	12
10. Revision History	13
10.1. Changes since -08 Draft	13
10.2. Changes since -07 Draft	13
10.3. Changes since -06 Draft	13
10.4. Changes since -05 Draft	14
10.5. Changes since -04 Draft	14
10.6. Changes since -03 Draft	14
10.7. Changes since -02 Draft	14
10.8. Changes since -01 Draft	14
10.9. Changes since -00 Draft	15
10.10. Changes since fenton-03 Draft	15
10.11. Changes Since -02 Draft	15
10.12. Changes Since -01 Draft	15
10.13. Changes Since -00 Draft	15
11. References	16
11.1. Normative References	16
11.2. Informative References	18
Appendix A. Examples	19

A.1. REQUIRETLS SMTP Option	19
A.2. TLS-Required Header Field	20
Author's Address	21

1. Introduction

The SMTP [RFC5321] STARTTLS service extension [RFC3207] provides a means by which an SMTP server and client can establish a Transport Layer Security (TLS) protected session for the transmission of email messages. By default, TLS is used only upon mutual agreement (successful negotiation) of STARTTLS between the client and server; if this is not possible, the message is sent without transport encryption. Furthermore, it is common practice for the client to negotiate TLS even if the SMTP server's certificate is invalid.

Policy mechanisms such as DANE [RFC7672] and MTA-STS [RFC8461] may impose requirements for the use of TLS for email destined for some domains. However, such policies do not allow the sender to specify which messages are more sensitive and require transport-level encryption, and which ones are less sensitive and ought to be relayed even if TLS cannot be negotiated successfully.

The default opportunistic nature of SMTP TLS enables several "on the wire" attacks on SMTP security between MTAs. These include passive eavesdropping on connections for which TLS is not used, interference in the SMTP protocol to prevent TLS from being negotiated (presumably accompanied by eavesdropping), and insertion of a man-in-the-middle attacker exploiting the lack of server authentication by the client. Attacks are described in more detail in the Security Considerations section of this document.

REQUIRETLS consists of two mechanisms: an SMTP service extension and a message header field. The service extension is used to specify that a given message sent during a particular session **MUST** be sent over a TLS-protected session with specified security characteristics. It also requires that the SMTP server advertise that it supports REQUIRETLS, in effect promising that it will honor the requirement to enforce TLS transmission and REQUIRETLS support for onward transmission of those messages.

The TLS-Required message header field is used to convey a request to ignore recipient-side policy mechanisms such as MTA-STS and DANE, thereby prioritizing delivery over ability to negotiate TLS. Unlike the service extension, the TLS-Required header field allows the message to transit through one or more MTAs that do not support REQUIRETLS.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The formal syntax uses the Augmented Backus-Naur Form (ABNF) [RFC5234] including the core rules defined in Appendix B of that document.

2. The REQUIRETLS Service Extension

1. The textual name of the extension is "Require TLS".
2. The EHLO keyword value associated with this extension is "REQUIRETLS".
3. No additional SMTP verbs are defined by this extension.
4. One optional parameter ("REQUIRETLS") is added to the MAIL FROM command by this extension. No value is associated with this parameter.
5. The maximum length of a MAIL FROM command line is increased by 11 octets by the possible addition of a space and the REQUIRETLS keyword.
6. One new SMTP status code is defined by this extension to convey an error condition resulting from failure of the client to send to a server not also supporting the REQUIRETLS extension.
7. The REQUIRETLS extension is valid for message relay [RFC5321], submission [RFC6409], and the Local Mail Transfer Protocol (LMTP) [RFC2033]
8. The ABNF syntax for the MAIL FROM parameter is as follows:

```
requiretls-param = "REQUIRETLS"  
                  ; where requiretls-param is an instance of an  
                  ; esmtp-param used in Mail-parameters in  
                  ; RFC 5321 Section 4.1.2. There is no esmtp-value  
                  ; associated with requiretls-param.
```

In order to specify REQUIRETLS treatment for a given message, the REQUIRETLS option is specified on the MAIL FROM command when that message is transmitted. This option MUST only be specified in the

context of an SMTP session meeting the security requirements of REQUIRETLS:

- o The session itself MUST employ TLS transmission.
- o If the SMTP server to which the message is being transmitted is identified through an MX record lookup, its name MUST be validated via a DNSSEC signature on the recipient domain's MX record, or the MX hostname MUST be validated by an MTA-STS policy as described in Section 4.1 of RFC 8461 [RFC8461]. DNSSEC is defined in RFC 4033 [RFC4033], RFC 4034 [RFC4034], and RFC 4035 [RFC4035].
- o The certificate presented by the SMTP server MUST either verify successfully in a trust chain leading to a certificate trusted by the SMTP client or it MUST verify successfully using DANE as specified in RFC 7672 [RFC7672]. For trust chains, the choice of trusted (root) certificates is at the discretion of the SMTP client.
- o Following the negotiation of STARTTLS, the SMTP server MUST advertise in the subsequent EHLO response that it supports REQUIRETLS.

3. The TLS-Required Header Field

One new message header field [RFC5322], TLS-Required, is defined by this specification. It is used for messages for which the originator requests that recipient TLS policy (including MTA-STS [RFC8461] and DANE [RFC7672]) be ignored. This might be done, for example, to report a misconfigured mail server, such as an expired TLS certificate.

The TLS-Required header field has a single REQUIRED parameter:

- o No - The SMTP client SHOULD attempt to send the message regardless of its ability to negotiate STARTTLS with the SMTP server, ignoring policy-based mechanisms (including MTA-STS and DANE), if any, asserted by the recipient domain. Nevertheless, the client SHOULD negotiate STARTTLS with the server if available.

More than one instance of the TLS-Required header field MUST NOT appear in a given message.

The ABNF syntax for the TLS-Required header field is as follows:


```
requiretls-field = "TLS-Required:" [FWS] "No" CRLF
                  ; where requiretls-field in an instance of an
                  ; optional-field defined in RFC 5322 Section
                  ; 3.6.8.
FWS = <as defined in RFC 5322>
CRLF = <as defined in RFC 5322>
```

4. REQUIRETLS Semantics

4.1. REQUIRETLS Receipt Requirements

Upon receipt of the REQUIRETLS option on a MAIL FROM command during the receipt of a message, an SMTP server MUST tag that message as needing REQUIRETLS handling.

Upon receipt of a message not specifying the REQUIRETLS option on its MAIL FROM command but containing the TLS-Required header field in its message header, an SMTP server implementing this specification MUST tag that message with the option specified in the TLS-Required header field. If the REQUIRETLS MAIL FROM parameter is specified, the TLS-Required header field MUST be ignored but MAY be included in onward relay of the message.

The manner in which the above tagging takes place is implementation-dependent. If the message is being locally aliased and redistributed to multiple addresses, all instances of the message MUST be tagged in the same manner.

4.2. REQUIRETLS Sender Requirements

4.2.1. Sending with TLS Required

When sending a message tagged as requiring TLS for which the MAIL FROM return-path is not empty (an empty MAIL FROM return-path indicating a bounce message), the sending (client) MTA MUST:

1. Look up the SMTP server to which the message is to be sent as described in [RFC5321] Section 5.1.
2. If the server lookup is accomplished via the recipient domain's MX record (the usual case) and is not accompanied by a valid DNSSEC signature, the client MUST also validate the SMTP server name using MTA-STX as described in RFC 8461 [RFC8461] Section 4.1.
3. Open an SMTP session with the peer SMTP server using the EHLO verb.

4. Establish a TLS-protected SMTP session with its peer SMTP server and authenticate the server's certificate as specified in [RFC6125] or [RFC7672] as applicable. The hostname from the MX record lookup (or the domain name in the absence of an MX record where an A record is used directly) MUST match the DNS-ID or CN-ID of the certificate presented by the server.
5. Ensure that the response to the subsequent EHLO following establishment of the TLS protection advertises the REQUIRETLS capability.

The SMTP client SHOULD follow the recommendations in [RFC7525] or its successor with respect to negotiation of the TLS session.

If any of the above steps fail, the client MUST issue a QUIT to the server and repeat steps 2-5 with each host on the recipient domain's list of MX hosts in an attempt to find a mail path that meets the sender's requirements. The client MAY send other, unprotected, messages to that server if it has any prior to issuing the QUIT. If there are no more MX hosts, the client MUST NOT transmit the message to the domain.

Following such a failure, the SMTP client MUST send a non-delivery notification to the reverse-path of the failed message as described in section 3.6 of [RFC5321]. The following status codes [RFC5248] SHOULD be used:

- o REQUIRETLS not supported by server: 5.7.YYY REQUIRETLS needed
- o Unable to establish TLS-protected SMTP session: 5.7.10 Encryption needed

Refer to Section 5 for further requirements regarding non-delivery messages.

If all REQUIRETLS requirements have been met, transmit the message, issuing the REQUIRETLS option on the MAIL FROM command with the required option(s), if any.

4.2.2. Sending with TLS Optional

Messages tagged TLS-Required: No are handled as follows. When sending such a message, the sending (client) MTA MUST:

- o Look up the SMTP server to which the message is to be sent as described in [RFC5321] Section 5.1.

- o Open an SMTP session with the peer SMTP server using the EHLO verb. Attempt to negotiate STARTTLS if possible, and follow any policy published by the recipient domain, but do not fail if this is unsuccessful.

Some SMTP servers may be configured to require STARTTLS connections as a matter of policy and not accept messages in the absence of STARTTLS. A non-delivery notification **MUST** be returned to the sender if message relay fails due to an inability to negotiate STARTTLS when required by the server.

Since messages tagged with TLS-Required: No will sometimes be sent to SMTP servers not supporting REQUIRETLS, that option will not be uniformly observed by all SMTP relay hops.

4.3. REQUIRETLS Submission

An MUA or other agent making the initial introduction of a message has the option to decide whether to require TLS. If TLS is to be required, it **MUST** do so by negotiating STARTTLS and REQUIRETLS and include the REQUIRETLS option on the MAIL FROM command, as is done for message relay.

When TLS is not to be required, the sender **MUST** include the TLS-Required header field in the message. SMTP servers implementing this specification **MUST** interpret this header field as described in Section 4.1.

In either case, the decision whether to specify REQUIRETLS **MAY** be done based on a user interface selection or based on a ruleset or other policy. The manner in which the decision to require TLS is made is implementation-dependent and is beyond the scope of this specification.

4.4. Delivery of REQUIRETLS messages

Messages are usually retrieved by end users using protocols other than SMTP such as IMAP [RFC3501], POP [RFC1939], or web mail systems. Mail delivery agents supporting the REQUIRETLS SMTP option **SHOULD** observe the guidelines in [RFC8314].

5. Non-delivery message handling

Non-delivery ("bounce") messages usually contain important metadata about the message to which they refer, including the original message header. They therefore **MUST** be protected in the same manner as the original message. All non-delivery messages resulting from messages with the REQUIRETLS SMTP option, whether resulting from a REQUIRETLS

error or some other, MUST also specify the REQUIRETLS SMTP option unless redacted as described below.

The path from the origination of an error bounce message back to the MAIL FROM address may not share the same REQUIRETLS support as the forward path. Therefore, users requiring TLS are advised to make sure that they are capable of receiving mail using REQUIRETLS as well. Otherwise, such non-delivery messages will be lost.

If a REQUIRETLS message is bounced, the server MUST behave as if RET=HDRS was present as described in [RFC3461]. If both RET=FULL and REQUIRETLS are present, the RET=FULL MUST be disregarded. The SMTP client for a REQUIRETLS bounce message uses an empty MAIL FROM return-path as required by [RFC5321]. When the MAIL FROM return-path is empty, the REQUIRETLS parameter SHOULD NOT cause a bounce message to be discarded even if the next-hop relay does not advertise REQUIRETLS.

Senders of messages requiring TLS are advised to consider the possibility that bounce messages will be lost as a result of REQUIRETLS return path failure, and that some information could be leaked if a bounce message is not able to be transmitted with REQUIRETLS.

6. Reorigination considerations

In a number of situations, a mediator [RFC5598] originates a new message as a result of an incoming message. These situations include, but are not limited to, mailing lists (including administrative traffic such as message approval requests), Sieve [RFC5228], "vacation" responders, and other filters to which incoming messages may be piped. These newly originated messages may essentially be copies of the incoming message, such as with a forwarding service or a mailing list expander. In other cases, such as with a vacation message or a delivery notification, they will be different but might contain parts of the original message or other information for which the original message sender wants to influence the requirement to use TLS transmission.

Mediators that reoriginate messages should apply REQUIRETLS requirements in incoming messages (both requiring TLS transmission and requesting that TLS not be required) to the reoriginated messages to the extent feasible. A limitation to this might be that for a message requiring TLS, redistribution to multiple addresses while retaining the TLS requirement could result in the message not being delivered to some of the intended recipients.

User-side mediators (such as use of Sieve rules on a user agent) typically do not have access to the SMTP details, and therefore may not be aware of the REQUIRETLS requirement on a delivered message. Recipients that expect sensitive traffic should avoid the use of user-side mediators. Alternatively, if operationally feasible (such as when forwarding to a specific, known address), they should apply REQUIRETLS to all reoriginated messages that do not contain the "TLS-Required: No" header field.

7. IANA Considerations

If published as an RFC, this draft requests the addition of the following keyword to the SMTP Service Extensions Registry [MailParams]:

Textual name:	Require TLS
EHLO keyword value:	REQUIRETLS
Syntax and parameters:	(no parameters)
Additional SMTP verbs:	none
MAIL and RCPT parameters:	REQUIRETLS parameter on MAIL
Behavior:	Use of the REQUIRETLS parameter on the MAIL verb causes that message to require the use of TLS and tagging with REQUIRETLS for all onward relay.
Command length increment:	11 characters

If published as an RFC, this draft requests the addition of an entry to the Simple Mail Transfer Protocol (SMTP) Enhanced Status Codes Registry [SMTPStatusCodes]:

Code:	5.7.YYY
Sample Text:	REQUIRETLS support required
Associated basic status code:	550
Description:	This indicates that the message was not able to be forwarded because it was received with a REQUIRETLS requirement and none of the SMTP servers to which the message should be forwarded provide this support.
Reference:	(this document)
Submitter:	J. Fenton
Change controller:	IESG

If published as an RFC, this draft requests the addition of an entry to the Permanent Message Header Field Names Registry [PermMessageHeaderFields]:

Header field name: TLS-Required
Applicable protocol: mail
Status: standard
Author/change controller: IETF
Specification document: (this document)

This section is to be updated for publication by the RFC Editor.

8. Security Considerations

The purpose of REQUIRETLS is to give the originator of a message control over the security of email they send, either by conveying an expectation that it will be transmitted in an encrypted form "over the wire" or explicitly that transport encryption is not required if it cannot be successfully negotiated.

The following considerations apply to the REQUIRETLS service extension but not the TLS-Required header field, since messages specifying the header field are less concerned with transport security.

8.1. Passive attacks

REQUIRETLS is generally effective against passive attackers who are merely trying to eavesdrop on an SMTP exchange between an SMTP client and server. This assumes, of course, the cryptographic integrity of the TLS connection being used.

8.2. Active attacks

Active attacks against TLS encrypted SMTP connections can take many forms. One such attack is to interfere in the negotiation by changing the STARTTLS command to something illegal such as XXXXXXXX. This causes TLS negotiation to fail and messages to be sent in the clear, where they can be intercepted. REQUIRETLS detects the failure of STARTTLS and declines to send the message rather than send it insecurely.

A second form of attack is a man-in-the-middle attack where the attacker terminates the TLS connection rather than the intended SMTP server. This is possible when, as is commonly the case, the SMTP client either does not verify the server's certificate or establishes the connection even when the verification fails. REQUIRETLS requires successful certificate validation before sending the message.

Another active attack involves the spoofing of DNS MX records of the recipient domain. An attacker having this capability could potentially cause the message to be redirected to a mail server under

the attacker's own control, which would presumably have a valid certificate. REQUIRETLS requires that the recipient domain's MX record lookup be validated either using DNSSEC or via a published MTA-STS policy that specifies the acceptable SMTP server hostname(s) for the recipient domain.

8.3. Bad Actor MTAs

A bad-actor MTA along the message transmission path could misrepresent its support of REQUIRETLS and/or actively strip REQUIRETLS tags from messages it handles. However, since intermediate MTAs are already trusted with the cleartext of messages they handle, and are not part of the threat model for transport-layer security, they are also not part of the threat model for REQUIRETLS.

It should be reemphasized that since SMTP TLS is a transport-layer security protocol, messages sent using REQUIRETLS are not encrypted end-to-end and are visible to MTAs that are part of the message delivery path. Messages containing sensitive information that MTAs should not have access to MUST be sent using end-to-end content encryption such as OpenPGP [RFC4880] or S/MIME [RFC8551].

8.4. Policy Conflicts

In some cases, the use of the TLS-Required header field may conflict with a recipient domain policy expressed through the DANE [RFC7672] or MTA-STS [RFC8461] protocols. Although these protocols encourage the use of TLS transport by advertising availability of TLS, the use of "TLS-Required: No" header field represents an explicit decision on the part of the sender not to require the use of TLS, such as to overcome a configuration error. The recipient domain has the ultimate ability to require TLS by not accepting messages when STARTTLS has not been negotiated; otherwise, "TLS-Required: No" is effectively directing the client MTA to behave as if it does not support DANE nor MTA-STS.

9. Acknowledgements

The author would like to acknowledge many helpful suggestions on the ietf-smtp and uta mailing lists, in particular those of Viktor Dukhovni, Chris Newman, Tony Finch, Jeremy Harris, Arvel Hathcock, John Klensin, Barry Leiba, John Levine, Rolf Sonneveld, and Per Thorsheim.

10. Revision History

To be removed by RFC Editor upon publication as an RFC.

10.1. Changes since -08 Draft

Additional changes in response to IESG review:

- o Unify wording describing TLS-Required in Appendix A.2.
- o Add specifics on verification of mail server hostnames with certificates.
- o Wording tweak in 4.3 to emphasize optional nature of REQUIRETLS.
- o Update S/MIME reference from RFC 5751 to 8551

10.2. Changes since -07 Draft

Changes in response to IESG review and IETF Last Call comments:

- o Change associated status code for 5.7.YYY from 530 to 550.
- o Correct textual name of extension in IANA Considerations for consistency with the rest of the document.
- o Remove special handling of bounce messages in Section 4.1.
- o Change name of header field from RequireTLS to TLS-Required and make capitalization of parameter consistent.
- o Remove mention of transforming RET=FULL to RET=HDRS on relay in Section 5.
- o Replace Section 6 dealing with mailing lists with a more general section on reorigination by mediators.
- o Add security considerations section on policy conflicts.

10.3. Changes since -06 Draft

Various changes in response to AD review:

- o Reference RFC 7525 for TLS negotiation recommendations.
- o Make reference to requested 5.7.YYY error code consistent.
- o Clarify applicability to LMTP and submission.

- o Provide ABNF for syntax of SMTP option and header field and examples in Appendix A.
- o Correct use of normative language in Section 5.
- o Clarify case where REQUIRETLS option is used on bounce messages.
- o Improve Security Requirements wording to be inclusive of both SMTP option and header field.

10.4. Changes since -05 Draft

Corrected IANA Permanent Message Header Fields Registry request.

10.5. Changes since -04 Draft

Require validation of SMTP server hostname via DNSSEC or MTA-STS policy when TLS is required.

10.6. Changes since -03 Draft

Working Group Last Call changes, including:

- o Correct reference for SMTP DANE
- o Clarify that RequireTLS: NO applies to both MTA-STS and DANE policies
- o Correct newly-defined status codes
- o Update MTA-STS references to RFC

10.7. Changes since -02 Draft

- o More complete documentation for IANA registration requests.
- o Changed bounce handling to use RET parameters of [RFC3461], along with slightly more liberal transmission of bounces even if REQUIRETLS can't be negotiated.

10.8. Changes since -01 Draft

- o Converted DEEP references to RFC 8314.
- o Removed REQUIRETLS options: CHAIN, DANE, and DNSSEC.
- o Editorial corrections, notably making the header field name consistent (RequireTLS rather than Require-TLS).

10.9. Changes since -00 Draft

- o Created new header field, Require-TLS, for use by "NO" option.
- o Removed "NO" option from SMTP service extension.
- o Recommend DEEP requirements for delivery of messages requiring TLS.
- o Assorted copy edits

10.10. Changes since fenton-03 Draft

- o Wording improvements from Rolf Sonneveld review 22 July 2017
- o A few copy edits
- o Conversion from individual to UTA WG draft

10.11. Changes Since -02 Draft

- o Incorporation of "MAY TLS" functionality as REQUIRETLS=NO per suggestion on UTA WG mailing list.
- o Additional guidance on bounce messages

10.12. Changes Since -01 Draft

- o Specified retries when multiple MX hosts exist for a given domain.
- o Clarified generation of non-delivery messages
- o Specified requirements for application of REQUIRETLS to mail forwarders and mailing lists.
- o Clarified DNSSEC requirements to include MX lookup only.
- o Corrected terminology regarding message retrieval vs. delivery.
- o Changed category to standards track.

10.13. Changes Since -00 Draft

- o Conversion of REQUIRETLS from an SMTP verb to a MAIL FROM parameter to better associate REQUIRETLS requirements with transmission of individual messages.

- o Addition of an option to require DNSSEC lookup of the remote mail server, since this affects the common name of the certificate that is presented.
- o Clarified the wording to more clearly state that TLS sessions must be established and not simply that STARTTLS is negotiated.
- o Introduced need for minimum encryption standards (key lengths and algorithms)
- o Substantially rewritten Security Considerations section

11. References

11.1. Normative References

[MailParams]

Internet Assigned Numbers Authority (IANA), "IANA Mail Parameters", 2007,
<<http://www.iana.org/assignments/mail-parameters>>.

[PermMessageHeaderFields]

Internet Assigned Numbers Authority (IANA), "Permanent Message Header Field Names Registry", 2004,
<<https://www.iana.org/assignments/message-headers/message-headers.xhtml#perm-headers>>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997,
<<https://www.rfc-editor.org/info/rfc2119>>.

[RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over Transport Layer Security", RFC 3207, DOI 10.17487/RFC3207, February 2002, <<https://www.rfc-editor.org/info/rfc3207>>.

[RFC3461] Moore, K., "Simple Mail Transfer Protocol (SMTP) Service Extension for Delivery Status Notifications (DSNs)", RFC 3461, DOI 10.17487/RFC3461, January 2003,
<<https://www.rfc-editor.org/info/rfc3461>>.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005,
<<https://www.rfc-editor.org/info/rfc4033>>.

- [RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <<https://www.rfc-editor.org/info/rfc4034>>.
- [RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <<https://www.rfc-editor.org/info/rfc4035>>.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/info/rfc5234>>.
- [RFC5248] Hansen, T. and J. Klensin, "A Registry for SMTP Enhanced Mail System Status Codes", BCP 138, RFC 5248, DOI 10.17487/RFC5248, June 2008, <<https://www.rfc-editor.org/info/rfc5248>>.
- [RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008, <<https://www.rfc-editor.org/info/rfc5321>>.
- [RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <<https://www.rfc-editor.org/info/rfc5322>>.
- [RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, DOI 10.17487/RFC6125, March 2011, <<https://www.rfc-editor.org/info/rfc6125>>.
- [RFC7525] Sheffer, Y., Holz, R., and P. Saint-Andre, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 7525, DOI 10.17487/RFC7525, May 2015, <<https://www.rfc-editor.org/info/rfc7525>>.
- [RFC7672] Dukhovni, V. and W. Hardaker, "SMTP Security via Opportunistic DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS)", RFC 7672, DOI 10.17487/RFC7672, October 2015, <<https://www.rfc-editor.org/info/rfc7672>>.

- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8314] Moore, K. and C. Newman, "Cleartext Considered Obsolete: Use of Transport Layer Security (TLS) for Email Submission and Access", RFC 8314, DOI 10.17487/RFC8314, January 2018, <<https://www.rfc-editor.org/info/rfc8314>>.
- [RFC8461] Margolis, D., Risher, M., Ramakrishnan, B., Brotman, A., and J. Jones, "SMTP MTA Strict Transport Security (MTA-STS)", RFC 8461, DOI 10.17487/RFC8461, September 2018, <<https://www.rfc-editor.org/info/rfc8461>>.
- [SMTPStatusCodes]
Internet Assigned Numbers Authority (IANA), "Simple Mail Transfer Protocol (SMTP) Enhanced Status Codes Registry", 2008, <<http://www.iana.org/assignments/smtp-enhanced-status-codes>>.

11.2. Informative References

- [RFC1939] Myers, J. and M. Rose, "Post Office Protocol - Version 3", STD 53, RFC 1939, DOI 10.17487/RFC1939, May 1996, <<https://www.rfc-editor.org/info/rfc1939>>.
- [RFC2033] Myers, J., "Local Mail Transfer Protocol", RFC 2033, DOI 10.17487/RFC2033, October 1996, <<https://www.rfc-editor.org/info/rfc2033>>.
- [RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, DOI 10.17487/RFC3501, March 2003, <<https://www.rfc-editor.org/info/rfc3501>>.
- [RFC4880] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, DOI 10.17487/RFC4880, November 2007, <<https://www.rfc-editor.org/info/rfc4880>>.
- [RFC5228] Guenther, P., Ed. and T. Showalter, Ed., "Sieve: An Email Filtering Language", RFC 5228, DOI 10.17487/RFC5228, January 2008, <<https://www.rfc-editor.org/info/rfc5228>>.
- [RFC5598] Crocker, D., "Internet Mail Architecture", RFC 5598, DOI 10.17487/RFC5598, July 2009, <<https://www.rfc-editor.org/info/rfc5598>>.

- [RFC6409] Gellens, R. and J. Klensin, "Message Submission for Mail", STD 72, RFC 6409, DOI 10.17487/RFC6409, November 2011, <<https://www.rfc-editor.org/info/rfc6409>>.
- [RFC8551] Schaad, J., Ramsdell, B., and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification", RFC 8551, DOI 10.17487/RFC8551, April 2019, <<https://www.rfc-editor.org/info/rfc8551>>.

Appendix A. Examples

This section is informative.

A.1. REQUIRETLS SMTP Option

The TLS-Required SMTP option is used to express the intent of the sender that the associated message be relayed using TLS. In the following example, lines beginning with C: are transmitted from the SMTP client to the server, and lines beginning with S: are transmitted in the opposite direction.

```
S: 220 mail.example.net ESMTP
C: EHLO mail.example.org
S: 250-mail.example.net Hello example.org [192.0.2.1]
S: 250-SIZE 52428800
S: 250-8BITMIME
S: 250-PIPELINING
S: 250-STARTTLS
S: 250 HELP
C: STARTTLS
S: TLS go ahead
```

(at this point TLS negotiation takes place. The remainder of this session occurs within TLS.)

```
S: 220 mail.example.net ESMTP
C: EHLO mail.example.org
S: 250-mail.example.net Hello example.org [192.0.2.1]
S: 250-SIZE 52428800
S: 250-8BITMIME
S: 250-PIPELINING
S: 250-REQUIRETLS
S: 250 HELP
C: MAIL FROM:<roger@example.org> REQUIRETLS
S: 250 OK
C: RCPT TO:<editor@example.net>
S: 250 Accepted
C: DATA
S: 354 Enter message, ending with "." on a line by itself
```

(message follows)

```
C: .
S: 250 OK
C: QUIT
```

A.2. TLS-Required Header Field

The TLS-Required header field is used when the sender requests that the mail system not heed a default policy of the recipient domain requiring TLS. It might be used, for example, to allow problems with the recipient domain's TLS certificate to be reported:

From: Roger Reporter <roger@example.org>
To: Andy Admin <admin@example.com>
Subject: Certificate problem?
TLS-Required: No
Date: Fri, 18 Jan 2019 10:26:55 -0800
Message-ID: <5c421a6f79c0e_d153ff8286d45c468473@mail.example.org>

Andy, there seems to be a problem with the TLS certificate
on your mail server. Are you aware of this?

Roger

Author's Address

Jim Fenton
Altmode Networks
Los Altos, California 94024
USA

Email: fenton@bluepopcorn.net

Human Rights Protocol Considerations Research Group
Internet-Draft
Intended status: Informational
Expires: August 27, 2018

S. Bortzmeyer
AFNIC
N. ten Oever
University of Amsterdam
February 23, 2018

Anonymity, Human Rights and Internet Protocols
draft-irtf-hrhc-anonymity-00

Abstract

Anonymity is less discussed in the IETF than for instance security [RFC3552] or privacy [RFC6973]. This can be attributed to the fact anonymity is a hard technical problem or that anonymizing user data is not of specific market interest. It remains a fact that 'most internet users would like to be anonymous online at least occasionally' [Pew].

This document aims to break down the different meanings and implications of anonymity on a mediated computer network.

Status of This Memo

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

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

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

This Internet-Draft will expire on August 27, 2018.

Copyright Notice

Copyright (c) 2018 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 (<https://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. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Vocabulary Used	3
3. Should protocols promote anonymity?	4
4. Example of use cases	5
4.1. Simultaneous use	5
4.2. Successive use	5
4.3. Selective use	6
4.4. User analysis	6
5. Practical advices	6
5.1. Protocol developers	6
5.2. Protocol implementors	7
6. Open Questions	7
7. Security Considerations	7
8. IANA Considerations	7
9. Research Group Information	8
10. Objections against anonymity	8
11. References	8
11.1. Informative References	8
11.2. URIs	10
Authors' Addresses	11

1. Introduction

There seems to be a clear need for anonymity online in an environment where harassment on the Internet is on the increase [Pew2] and the UN Special Rapporteur for Freedom of Expression calls anonymity 'necessary for the exercise of the right to freedom of opinion and expression in the digital age' [UNHRC2015].

Nonetheless anonymity is not getting much discussion at the IETF, providing anonymity does not seem a (semi-)objective for many protocols, even though several documents contribute to improving anonymity such as [RFC7258], [RFC7626], [RFC7858].

There are initiatives on the Internet to improve end users anonymity, most notably [torproject], but these initiatives rely on adding encryption in the application layer.

This document aims to break down the different meanings and implications of anonymity on a mediated computer network and to see

whether (some parts of) anonymity should be taken into consideration in protocol development.

2. Vocabulary Used

Concepts in this draft currently strongly hinges on [AnonTerm]

Anonymity A state of an individual in which an observer or attacker cannot identify the individual within a set of other individuals (the anonymity set). [RFC6973]

Linkability Linkability of two or more items of interest (IOIs - Items Of Interest, e.g., subjects, messages, actions, ...) from an attacker's perspective means that within the system (comprising these and possibly other items), the attacker can sufficiently distinguish whether these IOIs are related or not. [AnonTerm]

Official identity Government-issued identity, as written on ID cards and passports. We don't use terms like "real names" since a choosen pseudonym, for instance, is not less real than a identity given at birth.

Pseudonymity Derived from pseudonym, a persistent identity which is not the same as the entity's given (or official) name. For all IETF protocols, pseudonimity is a given: protocols don't care whether the identity is an official one or not. Even if the protocol allows to use official identities (for instance in the From: header of an Internet email), it does not require it. But it should be noted that, if the user cannot create new pseudonyms easily, pseudonyms suffer from linkability. Unlinkability depends on this ability to create new pseudonyms gratis and at will (good examples are SSH keys or Bitcoin addresses). Easy creation will allow to have one pseudonym per use, thus defeating linkability.

Unlinkability Unlinkability of two or more items of interest (IOIs, e.g., subjects, messages, actions, ...) from an attacker's perspective means that within the system (comprising these and possibly other items), the attacker cannot sufficiently distinguish whether these IOIs are related or not. [AnonTerm]

Undetectability The impossibility of being noticed or discovered

Undetectability of an item of interest (IOI) from an attacker's perspective means that the attacker cannot sufficiently distinguish whether it exists or not [AnonTerm]

Unobservability

Unobservability of an item of interest (IOI) means:
undetectability of the IOI against all subjects uninvolved in it
and

anonymity of the subject(s) involved in the IOI even against the
other subject(s) involved in that IOI. [AnonTerm]

It should be noted that the word "anonymity" is both very loaded politically (witness all the headlines about the "darknet") and poorly understood. Most texts talking about anonymity actually refer to pseudonymity (for instance, when people say that "Bitcoin is anonymous"). This confusion is even in the example given in [RFC4949] definition of anonymity.

Anonymity is strongly linked to unlinkability: if your actions are linkable, it suffices that one of them is tied to your identity, and anonymity is over.

It should be noted that anonymity is not binary: there have been these recent years a lot of progress of desanonymisation techniques (see also [GDPR], article 26). Data is never fully "anonymous", it is only more or less anonymous. [RFC6235] [MITdeano] [Utexas] [Article29]

3. Should protocols promote anonymity?

The amount of data that is generated by and about individuals is growing exponentially. This can be attributed to the fact that an ever increasing number of actions is digitally mediated, and the increase of connected sensors in the every day environment. Even though these two causes do not fully fall within the scope of the IETF, there is a significant part of these two examples that do.

TODO add here more examples of the need to anonymity

With the increase of data there is also an increasing ability for third parties to analyze human behaviour. It should be noted that any data that could identify an individual is personally identifiable information (PII). This means that information which can be used to distinguish an individual from other individuals can be considered as personally identifiable information. The access and control of personally identifiable information by a third party is a (potential) liability for both the third party and the individual. This liability could for example translate into a physical risk for the individual or into a legal risk for the third party under information security and privacy laws.

Some network operators argue that without the opportunity to persistently identify individual users it becomes harder to thwart attacks and troubleshoot network issues. Whereas identification might be helpful to address issues in some cases, it poses an inherent threat to the anonymity of users. Not protecting the anonymity of users leads to a deterioration of the right to privacy, and the right to freedom of opinion and expression. There can be limitations the right to privacy and freedom of expression, but these should always be provided by law and necessary and proportionate to achieve one of a handful of legitimate objectives. It is clear that anonymity may make system and network administration different. To quote [RFC7824], "Those properties (stable and trackable IP addresses, derived from static identifiers) are convenient for system administrators". Here, there is a clear and fundamental tussle between the protection of the users and the ability of the system and network administrator to continue their work in the same way.

Anonymity will always be a balancing act between user protection (which requires a high level of anonymity) and other requirements for operations and research, such as routing information. Anonymity is by no means achieved by default in an online environment, nor has it been a strong consideration in protocol development in the development of the Internet. Increasing anonymity in the digital environment is not an easy task, exactly because the ubiquity of data that is generated and stored. But exactly the fact that we generate so much data urges us to address this issue.

4. Example of use cases

4.1. Simultaneous use

One user may use concurrently several identities, mixing them in operations, while wanting to keep them distinct. The protocol and its implementations should not preclude this use.

4.2. Successive use

One user may switch from one identity to another. In that case, it must be doable without a "bleedover" from the old identity to the new one.

One of the reasons to switch identities might be to make the relationship between this identity and another one (for instance the official one) more difficult. The longer you use a pseudonym, the more clues you give to someone who tries to unveil pseudonymity.

4.3. Selective use

A user might want to retain their anonymity to certain actors / protocols, but identified to others. Also, she may also wish to be identified for some operations but not always.

4.4. User analysis

A user might want to understand which other actors might (potentially) have which level of information about them. This conflicts of course with privacy because the user has to reveal who he is. Example: if a domain name registry does not publish the name of a registrant, the registrant cannot check if the person who did the registration indicated the name of their client, or their own name.

5. Practical advices

5.1. Protocol developers

First, the protocol should avoid to have mandatory persistent identifiers.

Even without persistent identifiers, anonymity could be broken by examining the patterns of access. If an user visits each morning the three same Web sites, always in the same order, it will be easy to identify them even without persistent identifier. Protocol designers should therefore ask themselves if patterns are easily visible, or obfuscated in some way.

If the protocol collects data and distributes it (see [RFC6235]), "anonymizing" the data is often suggested but it is notoriously hard. Do not think that just dropping the last byte of an IP address "anonymizes" data.

Pay attention to the fact that Internet actors do not all see the same thing. Consider the anonymity of the user with respect to:

- local network operator
- other networks you connect to
- your communications peer on the other end of the pipe
- intermediaries ([RFC6973])
- enablers ([RFC6973])

- someone who is in several roles, for instance a big state surveillance agency

5.2. Protocol implementors

Avoid adding options or configurations that create or might lead to patterns or regularities that are not explicitly required by the protocol.

An example is DHCP where sending a persistent identifier as the client name was not mandatory but, in practice, done by many implementations, before [RFC7844].

If an implementation allows for identity management, there should be a clear barrier between the identities to ensure that they cannot (easily) be associated with each other.

If there are anonymization option for the protocol, these should be enabled by default.

6. Open Questions

While analyzing protocols for their impact on users anonymity, would it make sense to ask the following questions:

1. How does the protocol impact pseudonymity? If the protocol limits the creation of new pseudonyms, it can limit their usefulness to "hide" an user's identity. For instance, IP addresses are pseudonyms but, since they are not under end users's control, they have strong linkability. That's why they are rightly regarded as personal identifiers [EUCourt]. On the other hand, Bitcoin addresses are pseudonyms with limited linkability, since the user can always create a lot of them.
2. Could there be more advice for protocol developers and implementers to improve anonymity? (Besides the ones in Section 5.)

7. Security Considerations

As this draft concerns a research document, there are no security considerations.

8. IANA Considerations

This document has no actions for IANA.

9. Research Group Information

The discussion list for the IRTF Human Rights Protocol Considerations proposed working group is located at the e-mail address hrpc@ietf.org [1]. Information on the group and information on how to subscribe to the list is at <https://www.irtf.org/mailman/listinfo/hrpc> [2]

Archives of the list can be found at: <https://www.irtf.org/mail-archive/web/hrpc/current/index.html> [3]

10. Objections against anonymity

TODO: should be turned into an appendix. This draft is about how to allow anonymity, not about how to fight it.

For a long time, there have been objections against anonymity. This document won't attempt to rebuke them all, since it is concerned about how to ensure that protocols allow anonymity. But it is interesting to keep in mind that protocols never forbid anonymity. If someones want his or her actions to be trackable, and under her or his official name, they can do so, by adding this information to their messages. In the same way, people are free not to engage with anonymous entities, in the same way that a SIP use, for instance, is free not to pick up a call if it comes from `sip:anonymous@anonymous.invalid`. This document is concerned about enabling anonymity, not about mandating it.

11. References

11.1. Informative References

[AnonTerm]

Pfitzmann, A. and M. Hansen, "A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management", 2010, <http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf>.

[Article29]

Article29, ., "Opinion 05/2014 on Anonymisation Techniques", 2014, <http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf>.

- [EUCourt] "EUCJ Case C-70/10: Scarlet Extended SA vs. Societe belge des auteurs, compositeurs et editeurs SCRL (SABAM)", 2011, <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:62010CJ0070:EN:HTML&lipi=urn%3Ali%3Apage%3Ad_flagship3_pulse_read%3BSFHas%2FXMRHeHVu46775ezw%3D%3D>.
- [GDPR] European Parliament and Council, ., "REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)", 2016, <<http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679>>.
- [MITdeano] de Montjoye, Y., Hidalgo, C., Verleysen, M., and V. Blondel, "Unique in the Crowd: The privacy bounds of human mobility", 2013, <<https://www.nature.com/articles/srep01376>>.
- [Pew] Rainie, L., Kiesler, S., Kang, R., and M. Madden, "Anonymity, Privacy, and Security Online", 2013, <<http://www.pewinternet.org/2013/09/05/anonymity-privacy-and-security-online/>>.
- [Pew2] Duggan, M., "Online Harassment", 2014, <<http://www.pewinternet.org/2014/10/22/online-harassment/>>.
- [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>>.
- [RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <<https://www.rfc-editor.org/info/rfc4949>>.
- [RFC6235] Boschi, E. and B. Trammell, "IP Flow Anonymization Support", RFC 6235, DOI 10.17487/RFC6235, May 2011, <<https://www.rfc-editor.org/info/rfc6235>>.
- [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>>.
- [RFC7626] Bortzmeyer, S., "DNS Privacy Considerations", RFC 7626, DOI 10.17487/RFC7626, August 2015, <<https://www.rfc-editor.org/info/rfc7626>>.
- [RFC7824] Krishnan, S., Mrugalski, T., and S. Jiang, "Privacy Considerations for DHCPv6", RFC 7824, DOI 10.17487/RFC7824, May 2016, <<https://www.rfc-editor.org/info/rfc7824>>.
- [RFC7844] Huitema, C., Mrugalski, T., and S. Krishnan, "Anonymity Profiles for DHCP Clients", RFC 7844, DOI 10.17487/RFC7844, May 2016, <<https://www.rfc-editor.org/info/rfc7844>>.
- [RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <<https://www.rfc-editor.org/info/rfc7858>>.
- [torproject] The Tor Project, ., "Tor Project - Anonymity Online", 2007, <<https://www.torproject.org/>>.
- [UNHRC2015] Kaye, D., "Anonymity, Privacy, and Security Online (A/HRC/29/32)", 2015, <http://www.ohchr.org/EN/HRBodies/HRC/RegularSessions/Session29/Documents/A.HRC.29.32_AEV.doc>.
- [Utexas] Narayanan, A. and V. Shmatikov, "Robust De-anonymization of Large Sparse Datasets", 2008, <http://www.cs.utexas.edu/~shmat/shmat_oak08netflix.pdf>.

11.2. URIs

- [1] <mailto:hrpc@ietf.org>
- [2] <https://www.irtf.org/mailman/listinfo/hrpc>
- [3] <https://www.irtf.org/mail-archive/web/hrpc/current/index.html>

Authors' Addresses

Stephane Bortzmeyer
AFNIC

EMail: bortzmeyer+ietf@nic.fr

Niels ten Oever
University of Amsterdam

EMail: mail@nielstenoever.net

Human Rights Protocol Considerations Research Group	G. Grover
Internet-Draft	Centre for Internet and Society
Updates: 8280 (if approved)	N. ten Oever
Intended status: Informational	University of Amsterdam
Expires: 29 September 2022	28 March 2022

Guidelines for Human Rights Protocol and Architecture Considerations
draft-irtf-hrpc-guidelines-13

Abstract

This document sets guidelines for human rights considerations for developers working on network protocols and architectures, similar to the work done on the guidelines for privacy considerations [RFC6973]. This is an updated version of the guidelines for human rights considerations in [RFC8280].

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

This informational document has consensus for publication from the Internet Research Task Force (IRTF) Human Right Protocol Considerations Research Group. It has been reviewed, tried, and tested by both by the research group as well as by researchers and practitioners from outside the research group. The research group acknowledges that the understanding of the impact of internet protocols and architecture on society is a developing practice and is a body of research that is still in development.

Status of This Memo

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

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

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

This Internet-Draft will expire on 29 September 2022.

Copyright Notice

Copyright (c) 2022 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 (<https://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. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Human rights threats	4
3. Conducting human rights reviews	5
3.1. Analyzing drafts based on guidelines for human rights considerations model	6
3.2. Analyzing drafts based on their perceived or speculated impact	6
3.3. Expert interviews	6
3.4. Interviews with impacted persons and communities	6
3.5. Tracing impacts of implementations	6
4. Guidelines for human rights considerations	7
4.1. Connectivity	7
4.2. Reliability	8
4.3. Content agnosticism	9
4.4. Localization	10
4.5. Internationalization	11
4.6. Open Standards	12
4.7. Heterogeneity Support	13
4.8. Integrity	14
4.9. Authenticity	15
4.10. Confidentiality	16
4.11. Security	17
4.12. Privacy	18
4.13. Pseudonymity	18
4.14. Anonymity	19
4.15. Censorship resistance	20
4.16. Outcome Transparency	21
4.17. Adaptability	22
4.18. Accessibility	23
4.19. Decentralization	24
4.20. Remedy	25
4.21. Misc. considerations	25

5. Document Status	26
6. Acknowledgements	26
7. Security Considerations	26
8. IANA Considerations	26
9. Research Group Information	27
10. Informative References	27
Authors' Addresses	33

1. Introduction

This document outlines a set of human rights protocol considerations for protocol developers. It provides questions engineers should ask themselves when developing or improving protocols if they want to understand how their decisions can potentially influence the exercise of human rights on the Internet. It should be noted that the impact of a protocol cannot solely be deduced from its design, but its usage and implementation should also be studied to form a full protocol human rights impact assessment.

The questions are based on the research performed by the Human Rights Protocol Considerations (hrpc) research group which has been documented before these considerations. The research establishes that human rights relate to standards and protocols, and offers a common vocabulary of technical concepts that influence human rights and how these technical concepts can be combined to ensure that the Internet remains an enabling environment for human rights. With this, the contours of a model for developing human rights protocol considerations has taken shape.

This document is an iteration of the guidelines that can be found in [RFC8280]. The methods for conducting human rights reviews (Section 3.2), and guidelines for human rights considerations (Section 3.3) in this document are being tested for relevance, accuracy, and validity. The understanding of what human rights are is based on the Universal Declaration of Human Rights [UDHR] and subsequent treaties that jointly form the body of international human rights law [UNHR].

This document does not provide a detailed taxonomy of the nature of (potential) human rights violations, whether direct or indirect, long-term or short-term, certain protocol choices might present. In part because this is highly context-dependent, and in part, because this document aims to provide a practical set of guidelines. However, further research in this field would definitely benefit developers and implementers.

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

This informational document has consensus for publication from the Internet Research Task Force (IRTF) Human Right Protocol Considerations Research Group. It has been reviewed, tried, and tested by both by the research group as well as by researchers and practitioners from outside the research group. The research group acknowledges that the understanding of the impact of internet protocols and architecture on society is a developing practice and is a body of research that is still in development.

2. Human rights threats

Threats to the exercise of human rights on the Internet come in many forms. Protocols and standards may harm or enable the right to freedom of expression, right to freedom of information, right to non-discrimination, right to equal protection, right to participate in cultural life, arts and science, right to freedom of assembly and association, right to privacy, and the right to security. An end-user who is denied access to certain services or content may be unable to disclose vital information about the malpractices of a government or other authority. A person whose communications are monitored may be prevented or dissuaded from exercising their right to freedom of association or participate in political processes [Penney]. In a worst-case scenario, protocols that leak information can lead to physical danger. A realistic example to consider is when individuals perceived as threats to the state are subjected to torture, extra-judicial killing or detention on the basis of information gathered by state agencies through the monitoring of network traffic.

This document presents several examples of how threats to human rights materialize on the Internet. This threat modeling is inspired by [RFC6973] Privacy Considerations for Internet Protocols, which is based on security threat analysis. This method is a work in progress and by no means a perfect solution for assessing human rights risks in Internet protocols and systems. Certain specific human rights threats are indirectly considered in Internet protocols as part of the security considerations [BCP72], but privacy considerations [RFC6973] or reviews, let alone human rights impact assessments of protocols are not standardized or implemented.

Many threats, enablers, and risks are linked to different rights. This is not surprising if one takes into account that human rights are interrelated, interdependent, and indivisible. Here however we're not discussing all human rights because not all human rights are relevant to ICTs in general and protocols and standards in particular [Bless]: "The main source of the values of human rights is the International Bill of Human Rights that is composed of the Universal Declaration of Human Rights [UDHR] along with the

International Covenant on Civil and Political Rights [ICCPR] and the International Covenant on Economic, Social and Cultural Rights [ICESCR]. In the light of several cases of Internet censorship, the Human Rights Council Resolution 20/8 was adopted in 2012, affirming that "the same rights that people have offline must also be protected online." [UNHRC2016] In 2015, the Charter of Human Rights and Principles for the Internet [IRP] was developed and released. According to these documents, some examples of human rights relevant for ICT systems are human dignity (Art. 1 UDHR), non-discrimination (Art. 2), rights to life, liberty and security (Art. 3), freedom of opinion and expression (Art. 19), freedom of assembly and association (Art. 20), rights to equal protection, legal remedy, fair trial, due process, presumed innocent (Art. 7-11), appropriate social and international order (Art. 28), participation in public affairs (Art. 21), participation in cultural life, protection of the moral and material interests resulting from any scientific, literary or artistic production of which [they are] the author (Art. 27), and privacy (Art. 12)." A partial catalog of human rights related to Information and Communications Technologies, including economic rights, can be found in [Hill2014].

This is by no means an attempt to exclude specific rights or prioritize some rights over others.

3. Conducting human rights reviews

Ideally, protocol developers and collaborators should incorporate human rights considerations into the design process itself (see Guidelines for human rights considerations). This section provides guidance on how to conduct a human rights review, i.e. gauge the impact or potential impact of a protocol or standard on human rights.

Human rights reviews can take place at different stages of the development process of an Internet-Draft. Generally speaking, it is easier to influence the development of a technology at earlier stages than at later stages. This does not mean that reviews at last-call are not relevant, but they are less likely to result in significant changes in the reviewed document.

Methods for analyzing technology for specific human rights impacts are still quite nascent. Currently, five methods have been explored by the Human Rights Review Team, often in conjunction with each other:

3.1. Analyzing drafts based on guidelines for human rights considerations model

This analysis of Internet-Drafts uses the model as described in section 3.3. The outlined categories and questions can be used to review an Internet-Draft. The advantage of this is that it provides a known overview, and document authors can go back to this document as well as [RFC8280] to understand the background and the context.

3.2. Analyzing drafts based on their perceived or speculated impact

When reviewing an Internet-Draft, specific human rights impacts can become apparent by doing a close reading of the draft and seeking to understand how it might affect networks or society. While less structured than the straight use of the human rights considerations model, this analysis may lead to new speculative understandings of links between human rights and protocols.

3.3. Expert interviews

Interviews with document authors, active members of the Working Group, or experts in the field can help explore the characteristics of the protocol and its effects. There are two main advantages to this approach: one the one hand, it allows the reviewer to gain a deeper understanding of the (intended) workings of the protocol; on the other hand, it also allows for the reviewer to start a discussion with experts or even document authors, which can help the review gain traction when it is published.

3.4. Interviews with impacted persons and communities

Protocols impact users of the Internet. Interviews can help the reviewer understand how protocols affect the people that use the protocols. Since human rights are best understood from the perspective of the rights-holder, this approach will improve the understanding of the real world effects of the technology. At the same time, it can be hard to attribute specific changes to a particular protocol, this is of course even harder when a protocol has not been (widely) deployed.

3.5. Tracing impacts of implementations

The reality of deployed protocols can be at odds with the expectations during the protocol design and development phase [RFC8980]. When a specification already has associated running code, the code can be analyzed either in an experimental setting or on the Internet where its impact can be observed. In contrast to reviewing the draft text, this approach can allow the reviewer to understand

how the specifications works in practice, and potentially what unknown or unexpected effects the technology has.

4. Guidelines for human rights considerations

This section provides guidance for document authors in the form of a questionnaire about protocols and how technical decisions can shape the exercise of human rights. The questionnaire may be useful at any point in the design process, particularly after the document authors have developed a high-level protocol model as described in [RFC4101]. These guidelines do not seek to replace any existing referenced specifications, but rather contribute to them and look at the design process from a human rights perspective.

Protocols and Internet Standards might benefit from a documented discussion of potential human rights risks arising from potential misapplications of the protocol or technology described in the RFC. This might be coupled with an Applicability Statement for that RFC.

Note that the guidance provided in this section does not recommend specific practices. The range of protocols developed in the IETF is too broad to make recommendations about particular uses of data or how human rights might be balanced against other design goals. However, by carefully considering the answers to the following questions, document authors should be able to produce a comprehensive analysis that can serve as the basis for discussion on whether the protocol adequately takes specific human rights threats into account. This guidance is meant to help the thought process of a human rights analysis; it does not provide specific directions for how to write a human rights considerations section (following the example set in [RFC6973]).

In considering these questions, authors will need to be aware of the potential of technical advances or the passage of time to undermine protections. In general, considerations of rights are likely to be more effective if they are considered given a purpose and specific use cases, rather than as abstract absolute goals.

Also note that while the section uses the word, 'protocol', the principles identified in these questions may be applicable to other types of solutions (extensions to existing protocols, architecture for solutions to specific problems, etc.).

4.1. Connectivity

Question(s): Does your protocol add application-specific functions to intermediary nodes? Could this functionality be added to end nodes instead of intermediary nodes?

Is your protocol optimized for low bandwidth and high latency connections? Could your protocol also be developed in a stateless manner?

Explanation: The end-to-end principle [Saltzer] holds that certain functions can and should be performed at 'ends' of the network. [RFC1958] states "that in very general terms, the community believes that the goal is connectivity [...] and the intelligence is end to end rather than hidden in the network." Generally speaking, it is easier to attain reliability of data transmissions with computation at endpoints rather than at intermediary nodes.

Also considering the fact that network quality and conditions vary across geography and time, it is also important to design protocols such that they are reliable even on low bandwidth and high latency connections.

Example: Encrypting connections, like done with HTTPS, can add a significant network overhead and consequently make web resources less accessible to those with low bandwidth and/or high latency connections. [HTTPS-REL] Encrypting traffic is a net positive for privacy and security, and thus protocol designers can acknowledge the tradeoffs of connectivity made by such decisions.

Impacts:

- * Right to freedom of expression
- * Right to freedom of assembly and association

4.2. Reliability

Question(s): Is your protocol fault tolerant? Does it downgrade gracefully, i.e. with mechanisms for fallback and/or notice? Can your protocol resist malicious degradation attempts? Do you have a documented way to announce degradation? Do you have measures in place for recovery or partial healing from failure? Can your protocol maintain dependability and performance in the face of unanticipated changes or circumstances?

Explanation: Reliability and resiliency ensures that a protocol will execute its function consistently and error resistant as described, and function without unexpected result. Measures for reliability in protocols assure users that their intended communication was successfully executed.

A system that is reliable degrades gracefully and will have a documented way to announce degradation. It also has mechanisms to recover from failure gracefully, and if applicable, allow for partial healing.

It is important here to draw a distinction between random degradation and malicious degradation. Many current attacks against TLS, for example, exploit TLS' ability to gracefully downgrade to non-secure cipher suites - from a functional perspective, this is useful; from a security perspective, this can be disastrous. As with confidentiality, the growth of the Internet and fostering innovation in services depends on users having confidence and trust [RFC3724] in the network. For reliability, it is necessary that services notify the users if a delivery fails. In the case of real-time systems in addition to the reliable delivery the protocol needs to safeguard timeliness.

Example: In the modern IP stack structure, a reliable transport layer requires an indication that transport processing has successfully completed, such as given by TCP's ACK message [RFC0793], and not simply an indication from the IP layer that the packet arrived. Similarly, an application layer protocol may require an application-specific acknowledgment that contains, among other things, a status code indicating the disposition of the request (See [RFC3724]).

Impacts:

- * Right to freedom of expression
- * Right to security

4.3. Content agnosticism

Question(s): If your protocol impacts packet handling, does it use user data (packet data that is not included in the header)? Is it making decisions based on the payload of the packet? Does your protocol prioritize certain content or services over others in the routing process? Is the protocol transparent about the prioritization that is made (if any)?

Explanation: Content agnosticism refers to the notion that network traffic is treated identically regardless of payload, with some exceptions where it comes to effective traffic handling, for instance where it comes to delay-tolerant or delay-sensitive packets, based on the header. If there is any prioritization based on the content or metadata of the protocol, the protocol should be transparent about such information and reasons thereof.

Example: Content agnosticism prevents payload-based discrimination against packets. This is important because changes to this principle can lead to a two-tiered Internet, where certain packets are prioritized over others on the basis of their content. Effectively this would mean that although all users are entitled to receive their packets at a certain speed, some users become more equal than others.

Impacts:

- * Right to freedom of expression
- * Right to non-discrimination
- * Right to equal protection

4.4. Localization

Question(s): Does your protocol uphold the standards of internationalization? Have you made any concrete steps towards localizing your protocol for relevant audiences?

Explanation: Localization refers to the adaptation of a product, application or document content to meet the language, cultural and other requirements of a specific target market (a locale) [W3Cil18nDef]. For our purposes, it can be described as the practice of translating an implementation to make it functional in a specific language or for users in a specific locale (see Internationalization).

Example: The Internet is a global medium, but many of its protocols and products are developed with a certain audience in mind, that often share particular characteristics like knowing how to read and write in ASCII and knowing English. This limits the ability of a large part of the world's online population from using the Internet in a way that is culturally and linguistically accessible. An example of a protocol that has taken into account the view that individuals like to have access to data in their native language can be found in [RFC5646]. This protocol labels the information content with an identifier for the language in which it is written. And this allows information to be presented in more than one language.

Impacts:

- * Right to non-discrimination
- * Right to participate in cultural life, arts and science
- * Right to freedom of expression

4.5. Internationalization

Question(s): Does your protocol or specification define text string elements, in the payload or headers, that have to be understood or entered by humans? Does your specification allow Unicode? If so, do you accept texts in one charset (which must be UTF-8), or several (which is dangerous for interoperability)? If character sets or encodings other than UTF-8 are allowed, does your specification mandate a proper tagging of the charset? Did you have a look at [RFC6365]?

Explanation: Internationalization refers to the practice of making protocols, standards, and implementations usable in different languages and scripts (see Localization). In the IETF, internationalization means to add or improve the handling of non-ASCII text in a protocol. [RFC6365] A different perspective, more appropriate to protocols that are designed for global use from the beginning, is the definition used by W3C:

"Internationalization is the design and development of a product, application or document content that enables easy localization for target audiences that vary in culture, region, or language." {{W3Cil18nDef}}

Many protocols that handle text only handle one charset (US-ASCII), or leave the question of what coded character set and encoding are used up to local guesswork (which leads, of course, to interoperability problems). If multiple charsets are permitted, they must be explicitly identified [RFC2277]. Adding non-ASCII text to a protocol allows the protocol to handle more scripts, hopefully representing users across the world. In today's world, that is normally best accomplished by allowing Unicode encoded in UTF-8 only.

In the current IETF policy [RFC2277], internationalization is aimed at user-facing strings, not protocol elements, such as the verbs used by some text-based protocols. (Do note that some strings are both content and protocol elements, such as identifiers.) Given the IETF's mission to make the Internet a global network of networks, [RFC3935] developers should ensure that protocols work with languages apart from English and character sets apart from Latin characters. It is therefore crucial that at the very least, the content carried by the protocol can be in any script, and that all scripts are treated equally.

Example: See localization

Impacts:

- * Right to freedom of expression
- * Right to political participation
- * Right to participate in cultural life, arts and science

4.6. Open Standards

Question(s): Is your protocol fully documented in a way that it could be easily implemented, improved, built upon and/or further developed? Do you depend on proprietary code for the implementation, running or further development of your protocol? Does your protocol favor a particular proprietary specification over technically-equivalent competing specification(s), for instance by making any incorporated vendor specification "required" or "recommended" [RFC2026]? Do you normatively reference another standard that is not available without cost (and could you do without it)? Are you aware of any patents that would prevent your standard from being fully implemented [RFC8179] [RFC6701]?

Explanation: The Internet was able to be developed into the global network of networks because of the existence of open, non-proprietary standards [Zittrain]. They are crucial for enabling interoperability. Yet, open standards are not explicitly defined within the IETF. On the subject, [RFC2026] states: "Various national and international standards bodies, such as ANSI, ISO, IEEE, and ITU-T, develop a variety of protocol and service specifications that are similar to Technical Specifications defined at the IETF. National and international groups also publish "implementors' agreements" that are analogous to Applicability Statements, capturing a body of implementation-specific detail concerned with the practical application of their standards. All of these are considered to be "open external standards" for the purposes of the Internet Standards Process." Similarly, [RFC3935] does not define open standards but does emphasize the importance of an "open process", i.e. "any interested person can participate in the work, know what is being decided, and make his or her voice heard on the issue."

Open standards (and open source software) allow users to glean information about how the tools they are using work, including the tools' security and privacy properties. They additionally allow for permissionless innovation, which is important to maintain the freedom and ability to freely create and deploy new protocols on top of the communications constructs that currently exist. It is at the heart of the Internet as we know it, and to maintain its fundamentally open nature, we need to be mindful of the need for developing open standards.

All standards that need to be normatively implemented should be freely available and with reasonable protection for patent infringement claims, so it can also be implemented in open source or free software. Patents have often held back open standardization or been used against those deploying open standards, particularly in the domain of cryptography [newegg]. An exemption of this is sometimes made when a protocol is standardized that normatively relies on specifications produced by others SDOs that are not freely available. Patents in open standards or in normative references to other standards should have a patent disclosure [notewell], royalty-free licensing [patentpolicy], or some other form of fair, reasonable and non-discriminatory terms.

Example: [RFC6108] describes a system for providing critical end-user notifications to web browsers, which has been deployed by Comcast, an Internet Service Provider (ISP). Such a notification system is being used to provide near-immediate notifications to customers, such as to warn them that their traffic exhibits patterns that are indicative of malware or virus infection. There are other proprietary systems that can perform such notifications, but those systems utilize Deep Packet Inspection (DPI) technology. In contrast, that document describes a system that does not rely upon DPI, and is instead based on open IETF standards and open source applications.

Impacts:

- * Right to freedom of expression
- * Right to participate in cultural life, arts and science

4.7. Heterogeneity Support

Question(s): Does your protocol support heterogeneity by design? Does your protocol allow for multiple types of hardware? Does your protocol allow for multiple types of application protocols? Is your protocol liberal in what it receives and handles? Will it remain usable and open if the context changes?

Explanation: The Internet is characterized by heterogeneity on many levels: devices and nodes, router scheduling algorithms and queue management mechanisms, routing protocols, levels of multiplexing, protocol versions and implementations, underlying link layers (e.g., point-to-point, multi-access links, wireless, FDDI, etc.), in the traffic mix and in the levels of congestion at different times and places. Moreover, as the Internet is composed of autonomous organizations and Internet service providers, each with their own separate policy concerns, there is a large heterogeneity of administrative domains and pricing structures. As a result, the heterogeneity principle proposed in [RFC1958] needs to be supported by design [FIArch].

Heterogeneity support in protocols can thus enable a wide range of devices and (by extension) users to participate on the network.

Example: Heterogeneity is inevitable and needs be supported by design. Multiple types of hardware must be allowed for, e.g. transmission speeds differing by at least 7 orders of magnitude, various computer word lengths, and hosts ranging from memory-starved microprocessors up to massively parallel supercomputers. Multiple types of application protocols must be allowed for, ranging from the simplest such as remote login up to the most complex such as commit protocols for distributed databases. [RFC1958].

Impacts:

- * Right to freedom of expression
- * Right to political participation

4.8. Integrity

Question(s): Does your protocol maintain, assure and/or verify the accuracy of payload data? Does your protocol maintain and assure the consistency of data? Does your protocol in any way allow for the data to be (intentionally or unintentionally) altered?

Explanation: Integrity refers to the maintenance and assurance of the accuracy and consistency of data to ensure it has not been (intentionally or unintentionally) altered.

Example: Integrity verification of data is important to prevent vulnerabilities and attacks from on-path attackers. These attacks happen when a third party (often for malicious reasons) intercepts a communication between two parties, inserting themselves in the middle changing the content of the data. In practice this looks as follows:

Alice wants to communicate with Bob. Corinne forges and sends a message to Bob, impersonating Alice. Bob cannot see the data from Alice was altered by Corinne. Corinne intercepts and alters the communication as it is sent between Alice and Bob. Corinne is able to control the communication content.

Impacts:

- * Right to freedom of expression
- * Right to security

4.9. Authenticity

Question(s): Do you have sufficient measures to confirm the truth of an attribute of a single piece of data or entity? Can the attributes get garbled along the way (see security)? If relevant, have you implemented IPsec, DNSsec, HTTPS and other Standard Security Best Practices?

Explanation: Authenticity ensures that data does indeed come from the source it claims to come from. This is important to prevent certain attacks or unauthorized access and use of data.

At the same time, authentication should not be used as a way to prevent heterogeneity support, as is often done for vendor lock-in or digital rights management.

Example: Authentication of data is important to prevent vulnerabilities, and attacks from on-path attackers. These attacks happen when a third party (often for malicious reasons) intercepts a communication between two parties, inserting themselves in the middle and posing as both parties. In practice this looks as follows:

Alice wants to communicate with Bob. Alice sends data to Bob. Corinne intercepts the data sent to Bob. Corinne reads (and potentially alters) the message to Bob. Bob cannot see the data did not come from Alice but from Corinne.

When there is proper authentication the scenario would be as follows:

Alice wants to communicate with Bob. Alice sends data to Bob. Corinne intercepts the data sent to Bob. Corinne reads and alters the message to Bob. Bob can see the data did not come from Alice.

Impacts:

- * Right to privacy

- * Right to freedom of expression

- * Right to security

4.10. Confidentiality

Question(s): Does this protocol expose the transmitted data over the wire? Does the protocol expose information related to identifiers or data? If so, does it do so to each other protocol entity (i.e., recipients, intermediaries, and enablers) [RFC6973]? What options exist for protocol implementers to choose to limit the information shared with each entity? What operational controls are available to limit the information shared with each entity?

What controls or consent mechanisms does the protocol define or require before personal data or identifiers are shared or exposed via the protocol? If no such mechanisms or controls are specified, is it expected that control and consent will be handled outside of the protocol?

Does the protocol provide ways for initiators to share different pieces of information with different recipients? If not, are there mechanisms that exist outside of the protocol to provide initiators with such control?

Does the protocol provide ways for initiators to limit the sharing or express individuals' preferences to recipients or intermediaries with regard to the collection, use, or disclosure of their personal data? If not, are there mechanisms that exist outside of the protocol to provide users with such control? Is it expected that users will have relationships that govern the use of the information (contractual or otherwise) with those who operate these intermediaries? Does the protocol prefer encryption over clear text operation?

Explanation: Confidentiality refers to keeping your data secret from unintended listeners [BCP72]. The growth of the Internet depends on users having confidence that the network protects their personal data [RFC1984]. The possibility of pervasive monitoring and surveillance undermines users' trust, and can be mitigated by ensuring confidentiality, i.e. passive attackers should gain little or no information from observation or inference of protocol activity. [RFC7258][RFC7624].

Example: Protocols that do not encrypt their payload make the entire content of the communication available to the idealized attacker along their path. Following the advice in [RFC3365], most such protocols have a secure variant that encrypts the payload for confidentiality, and these secure variants are seeing ever-wider

deployment. A noteworthy exception is DNS [RFC1035], as DNSSEC [RFC4033] does not have confidentiality as a requirement. This implies that, in the absence of the use of more recent standards like DNS over TLS [RFC7858] or DNS over HTTPS [RFC8484], all DNS queries and answers generated by the activities of any protocol are available to the attacker. When store-and-forward protocols are used (e.g., SMTP [RFC5321]), intermediaries leave this data subject to observation by an attacker that has compromised these intermediaries, unless the data is encrypted end-to-end by the application-layer protocol or the implementation uses an encrypted store for this data [RFC7624].

Impacts:

- * Right to privacy
- * Right to security

4.11. Security

Question(s): Did you have a look at Guidelines for Writing RFC Text on Security Considerations [BCP72]? Have you found any attacks that are somewhat related to your protocol/specification, yet considered out of scope of your document? Would these attacks be pertinent to the human rights enabling features of the Internet (as described throughout this document)?

Explanation: Security is not a single monolithic property of a protocol or system, but rather a series of related but somewhat independent properties. Not all of these properties are required for every application. Since communications are carried out by systems and access to systems is through communications channels, security goals obviously interlock, but they can also be independently provided. [BCP72].

Typically, any protocol operating on the internet can be the target of passive attacks (when the attacker can access and read packets on the network); active attacks (when an attacker is capable of writing information to the network packets). [BCP72]

Example: See [BCP72].

Impacts:

- * Right to freedom of expression
- * Right to freedom of assembly and association

- * Right to non-discrimination
- * Right to security

4.12. Privacy

Question(s): Did you have a look at the Guidelines in the Privacy Considerations for Internet Protocols [RFC6973] section 7? Does your protocol maintain the confidentiality of metadata? Could your protocol counter traffic analysis? Does your protocol adhere to data minimization principles? Does your document identify potentially sensitive data logged by your protocol and/or for how long that needs to be retained for technical reasons?

Explanation: Privacy refers to the right of an entity (normally a person), acting on its own behalf, to determine the degree to which it will interact with its environment, including the degree to which the entity is willing to share its personal information with others. [RFC4949]. If a protocol provides insufficient privacy protection it may have a negative impact on freedom of expression as users self-censor for fear of surveillance, or find themselves unable to express themselves freely.

Example: See [RFC6973]

Impacts:

- * Right to freedom of expression
- * Right to non-discrimination

4.13. Pseudonymity

Question(s): Does the protocol collect personally derived data? Does the protocol generate or process anything that can be, or be tightly correlated with, personally identifiable information? Does the protocol utilize data that is personally-derived, i.e. derived from the interaction of a single person, or their device or address? If yes, can the protocol be implemented in a way that does not rely on personally identifiable information? If not, does the specification describe how any such data be handled? Have you considered the Privacy Considerations for Internet Protocols [RFC6973], especially section 6.1.2?

Explanation: Pseudonymity means using a pseudonym instead of one's "real" name. There are many reasons for users to use pseudonyms, for instance to: hide their gender, protect themselves against harassment, protect their families' privacy, frankly discuss

sexuality, or develop an artistic or journalistic persona without repercussions from an employer, (potential) customers, or social surrounding. [geekfeminism] The difference between anonymity and pseudonymity is that a pseudonym often is persistent. "Pseudonymity is strengthened when less personal data can be linked to the pseudonym; when the same pseudonym is used less often and across fewer contexts; and when independently chosen pseudonyms are more frequently used for new actions (making them, from an observer's or attacker's perspective, unlinkable)." [RFC6973]

Pseudonymity - the ability to use a persistent identifier not linked to one's offline identity - is an important feature for many end-users, as it allows them different degrees of disguised identity and privacy online. This can allow an enabling environment for users to exercise other rights, including freedom of expression and political participation, without fear or direct identification or discrimination.

Example: Generally, pseudonymous identifiers cannot be simply reverse engineered. Some early approaches took approaches such as simple hashing of IP addresses, but these could then be simply reversed by generating a hash for each potential IP address and comparing it to the pseudonym.

Example: There are also efforts for application layer protocols, like Oblivious DNS Over HTTPS, [draft-pauly-dprive-oblivious-doh] that can separate identifiers from requests.

Impacts:

- * Right to non-discrimination
- * Right to freedom of expression
- * Right to political participation
- * Right to freedom of assembly and association

4.14. Anonymity

Question(s): Does your protocol make use of persistent identifiers? Can it be done without them? Did you have a look at the Privacy Considerations for Internet Protocols [RFC6973], especially section 6.1.1 of that document?

Explanation: Anonymity refers to the condition of an identity being unknown or concealed [RFC4949]. Even though full anonymity is hard to achieve, it is a non-binary concept. Making pervasive monitoring

and tracking harder is important for many users as well as for the IETF [RFC7258]. Achieving a higher level of anonymity is an important feature for many end-users, as it allows them different degrees of privacy online. Anonymity is an inherent part of the right to freedom of opinion and expression and the right to privacy. Avoid adding identifiers, options or configurations that create or might lead to patterns or regularities that are not explicitly required by the protocol.

If your protocol collects data and seeks to distribute it to more entities than the originally-intended recipients (see [RFC6235] as an example), you should anonymize the data, but keep in mind that "anonymizing" data is notoriously hard. For example, just dropping the last byte of an IP address does not "anonymize" data.

If your protocol allows for identity management, there should be a clear barrier between the identities to ensure that they cannot (easily) be associated with each other.

A protocol that uses data that could help identify a sender (items of interest) should be protected from third parties. For instance, if one wants to hide the source/destination IP addresses of a packet, the use of IPsec in tunneling mode (e.g., inside a virtual private network) can be helpful to protect from third parties likely to eavesdrop packets exchanged between the tunnel endpoints.

Example: An example is DHCP where sending a persistent identifier as the client name was not mandatory but, in practice, done by many implementations, before [RFC7844].

Impacts:

- * Right to non-discrimination
- * Right to political participation
- * Right to freedom of assembly and association
- * Right to security

4.15. Censorship resistance

Question(s): Can your protocol contribute to filtering? Could be implemented to censor data or services? Could it be designed to ensure this doesn't happen? Does your protocol make it apparent or transparent when access to a resource is restricted and reasons therefor? Does your protocol introduce new identifiers or reuse existing identifiers (e.g. MAC addresses) that might be associated

with persons or content?

Explanation: Governments and service providers block or filter content or traffic, often without the knowledge of end-users. [RFC7754] See [draft-irtf-pearg-censorship] for a survey of censorship techniques employed across the world, which lays out protocol properties that have been exploited to censor access to information. Censorship resistance refers to the methods and measures to prevent Internet censorship.

Example: Identifiers of content exposed within a protocol might be used to facilitate censorship, as in the case of Application Layer based censorship, which affects protocols like HTTP. In HTTP, denial or restriction of access can be made apparent by the use of status code 451, which allows server operators to operate with greater transparency in circumstances where issues of law or public policy affect their operation [RFC7725].

If a protocol potentially enables censorship, protocol designers should strive towards creating error codes that capture different scenarios (blocked due to administrative policy, unavailable because of legal requirements, etc.) to minimize ambiguity for end-users.

In the development of the IPv6 protocol, it was discussed to embed a Media Access Control (MAC) address into unique IP addresses. This would make it possible for 'eavesdroppers and other information collectors to identify when different addresses used in different transactions actually correspond to the same node. This is why standardisation efforts like Privacy Extensions for Stateless Address Autoconfiguration in IPv6 [RFC4941] and MAC address randomization [draft-zuniga-mac-address-randomization] have been pursued.

Impacts:

- * Right to freedom of expression
- * Right to political participation
- * Right to participate in cultural life, arts, and science
- * Right to freedom of assembly and association

4.16. Outcome Transparency

Question(s): Are the effects of your protocol fully and easily comprehensible, including with respect to unintended consequences of protocol choices?

Explanation: Certain technical choices may have unintended consequences.

Example: Lack of authenticity may lead to lack of integrity and negative externalities, of which spam is an example. Lack of data that could be used for billing and accounting can lead to so-called "free" arrangements which obscure the actual costs and distribution of the costs, for example the barter arrangements that are commonly used for Internet interconnection; and the commercial exploitation of personal data for targeted advertising which is the most common funding model for the so-called "free" services such as search engines and social networks. Other unexpected outcomes might not be technical, but rather architectural, social or economical.

Impacts:

- * Right to freedom of expression
- * Right to privacy
- * Right to freedom of assembly and association
- * Right to access to information

4.17. Adaptability

Question(s): Is your protocol written in such a way that it would be easy for other protocols to be developed on top of it, or to interact with it? Does your protocol impact permissionless innovation? (See Open Standards)

Explanation: Adaptability is closely interrelated with permissionless innovation: both maintain the freedom and ability to freely create and deploy new protocols on top of the communications constructs that currently exist. It is at the heart of the Internet as we know it, and to maintain its fundamentally open nature, we need to be mindful of the impact of protocols on maintaining or reducing permissionless innovation to ensure the Internet can continue to develop.

Adaptability and permissionless innovation can be used to shape information networks as preferenced by groups of users. Furthermore, a precondition of adaptability is the ability of the people who can adapt the network to be able to know and understand the network. This is why adaptability and permissionless innovation are inherently connected to the right to education and the right to science as well as the right to freedom of assembly and association as well as the right to freedom of expression. Since it allows the users of the network to determine how the assemble, collaborate, and express themselves.

Example: WebRTC generates audio and/or video data. In order to ensure that WebRTC can be used in different locations by different parties, it is important that standard Javascript APIs are developed to support applications from different voice service providers. Multiple parties will have similar capabilities, in order to ensure that all parties can build upon existing standards these need to be adaptable, and allow for permissionless innovation.

Impacts:

- * Right to education
- * Right to science
- * Right to freedom of expression
- * Right to freedom of assembly and association

4.18. Accessibility

Question(s): Is your protocol designed to provide an enabling environment for all? Have you looked at the W3C Web Accessibility Initiative for examples and guidance?

Explanation: Sometimes in the design of protocols, websites, web technologies, or web tools, barriers are created that exclude people from using the Web. The Internet should be designed to work for all people, whatever their hardware, software, language, culture, location, or physical or mental ability. When the Internet technologies meet this goal, it will be accessible to people with a diverse range of hearing, movement, sight, and cognitive ability. [W3CAccessibility]

Example: The HTML protocol as defined in [HTML5] specifically requires that every image must have an alt attribute (with a few exceptions) to ensure images are accessible for people that cannot themselves decipher non-text content in web pages.

Another example is the works that is done in the AVT and AVTCORE working groups in the IETF that enable text conversation in multimedia, text telephony, wireless multimedia and video communications for sign language and lip-reading (ie. [RFC9071]).

Impacts:

- * Right to non-discrimination
- * Right to freedom of assembly and association
- * Right to education
- * Right to political participation

4.19. Decentralization

Question(s): Can your protocol be implemented without a single point of control? If applicable, can your protocol be deployed in a federated manner? Does your protocol create additional centralized points of control?

Explanation: Decentralization is one of the central technical concepts of the architecture of the Internet, and is embraced as such by the IETF [RFC3935]. It refers to the absence or minimization of centralized points of control, a feature that is assumed to make it easy for new users to join and new uses to unfold [Brown]. It also reduces issues surrounding single points of failure, and distributes the network such that it continues to function even if one or several nodes are disabled. With the commercialization of the Internet in the early 1990s, there has been a slow move away from decentralization, to the detriment of the technical benefits of having a decentralized Internet. For a more detailed discussion of this topic, please see [arkkoetal].

Example: The bits traveling the Internet are increasingly susceptible to monitoring and censorship, from both governments and Internet service providers, as well as third (malicious) parties. The ability to monitor and censor is further enabled by the increased centralization of the network that creates central infrastructure points that can be tapped into. The creation of peer-to-peer networks and the development of voice-over-IP protocols using peer-to-peer technology in combination with distributed hash table (DHT) for scalability are examples of how protocols can preserve decentralization [Pouwelse].

Impacts:

- * Right to freedom of expression
- * Right to freedom of assembly and association

4.20. Remedy

Question(s): Can your protocol facilitate a negatively impacted party's right to remedy without disproportionately impacting other parties' human rights, especially their right to privacy?

Explanation: Providing access to remedy by states and corporations is a part of the UN Guiding Principles on Business and Human Rights [UNGP]. Access to remedy may help victims of human rights violations in seeking justice, or allow law enforcement agencies to identify a possible violator. However, mechanisms in protocols that try to enable 'attribution' to individuals will impede the exercise of the right to privacy. The former Special Rapporteur for Freedom of Expression has also argued that anonymity is an inherent part of freedom of expression [Kaye]. Considering the potential adverse impact of attribution on the right to privacy and freedom of expression, enabling attribution on an individual level is most likely not consistent with human rights.

Example: Adding personal identifiable information to data streams might help in identifying a violator of human rights and provide access to remedy, but this would disproportionately affect all users right to privacy, anonymous expression, and association.

Impacts:

- * Right to remedy
- * Right to security
- * Right to privacy

4.21. Misc. considerations

Question(s): Have you considered potential negative consequences (individual or societal) that your protocol or document might have?

Explanation: Publication of a particular RFC under a certain status has consequences. Publication as an Internet Standard as part of the Standards Track may signal to implementers that the specification has a certain level of maturity, operational experience, and consensus. Similarly, publication of a specification as an experimental document as part of the non-standards track would signal to the community that the document "may be intended for eventual standardization but [may]

not yet [be] ready" for wide deployment. The extent of the deployment, and consequently its overall impact on end-users, may depend on the document status presented in the RFC. See [BCP9] and updates to it for a fuller explanation.

5. Document Status

This RG document is currently documenting best practices and guidelines for human rights reviews of network protocols, architectures and other Internet-Drafts and RFCs.

6. Acknowledgements

Thanks to:

- * Corinne Cath-Speth for work on [RFC8280].
- * Theresa Engelhard, Joe Hall, Avri Doria, Joey Salazar, Corinne Cath-Speth, Farzaneh Badii, Sandra Braman, Colin Perkins, John Curran, Eliot Lear, Mallory Knodel, and the hrpc list for reviews and suggestions.
- * Individuals who conducted human rights reviews for their work and feedback: Amelia Andersdotter, Beatrice Martini, Karan Saini and Shivan Kaul Sahib.

7. Security Considerations

Article three of the Universal Declaration of Human Rights reads: "Everyone has the right to life, liberty and security of person.". This article underlines the importance of security and its interrelation with human life and liberty, but since human rights are indivisible, interrelated and interdependent, security is also closely linked to other human rights and freedoms. This document seeks to strengthen human rights, freedoms, and security by relating and translating these concepts to concepts and practices as they are used in Internet protocol and architecture development. The aim of this is to secure human right and thereby improve the sustainability, usability, and effectiveness of the network. The document seeks to achieve this by providing guidelines as done in section three of this document.

8. IANA Considerations

This document has no actions for IANA.

9. Research Group Information

The discussion list for the IRTF Human Rights Protocol Considerations Research Group is located at the e-mail address hrpc@ietf.org (<mailto:hrpc@ietf.org>). Information on the group and information on how to subscribe to the list is at <https://www.irtf.org/mailman/listinfo/hrpc> (<https://www.irtf.org/mailman/listinfo/hrpc>)

Archives of the list can be found at: <https://www.irtf.org/mail-archive/web/hrpc/current/index.html> (<https://www.irtf.org/mail-archive/web/hrpc/current/index.html>)

10. Informative References

[arkkoetal]

Arkko, J., Trammell, B., Nottingham, M., Huitema, C., Thomson, M., Tantsure, J., and N. ten Oever, "Considerations on Internet Consolidation and the Internet Architecture", 2019, <https://datatracker.ietf.org/doc/html/draft-arkko-iab-internet-consolidation-02>.

[BCP72]

IETF, "Guidelines for Writing RFC Text on Security Considerations", 2003, <https://datatracker.ietf.org/doc/bcp72>.

[BCP9]

Bradner, S. and IETF, "The Internet Standards Process -- Revision 3", 1996, <https://datatracker.ietf.org/doc/rfc2026>.

[Bless]

Bless, R. and C. Orwat, "Values and Networks", 2015.

[Brown]

Brown, I. and M. Ziewitz, "A Prehistory of Internet Governance", Research Handbook on Governance of the Internet. Cheltenham, Edward Elgar. , 2013.

[draft-irtf-pearg-censorship]

Hall, J., Aaron, M., Adams, S., Jones, B., and N. Feamster, "A Survey of Worldwide Censorship Techniques", 2020, <https://tools.ietf.org/html/draft-irtf-pearg-censorship>.

[draft-pauly-dprive-oblivious-doh]

Kinnear, E., McManus, P., Pauly, T., Verma, T., and C.A. Wood, "Oblivious DNS Over HTTPS", 2022, <https://tools.ietf.org/html/draft-pauly-dprive-oblivious-doh>.

- [draft-zuniga-mac-address-randomization]
Zuniga, JC., Bernardos, CJ., and A. Andersdotter, "MAC address randomization", 2020,
<<https://tools.ietf.org/html/draft-irtf-pearg-censorship>>.
- [FIArch] "Future Internet Design Principles", January 2012,
<http://www.future-internet.eu/uploads/media/FIArch_Design_Principles_V1.0.pdf>.
- [geekfeminism]
Geek Feminism Wiki, "Pseudonymity", 2015,
<<http://geekfeminism.wikia.com/wiki/Pseudonymity>>.
- [Hill2014] Hill, R., "Partial Catalog of Human Rights Related to ICT Activities", 2014,
<<http://www.apig.ch/UNIGE%20Catalog.pdf>>.
- [HTML5] W3C, "HTML5", 2014, <<https://www.w3.org/TR/html5/>>.
- [HTTPS-REL]
Meyer, E., "Securing Web Sites Made Them Less Accessible", 2018, <<https://meyerweb.com/eric/thoughts/2018/08/07/securing-sites-made-them-less-accessible/>>.
- [ICCPR] United Nations General Assembly, "International Covenant on Civil and Political Rights", 1976,
<<http://www.ohchr.org/EN/ProfessionalInterest/Pages/CCPR.aspx>>.
- [ICESCR] United Nations General Assembly, "International Covenant on Economic, Social and Cultural Rights", 1966,
<<http://www.ohchr.org/EN/ProfessionalInterest/Pages/CESCR.aspx>>.
- [IRP] Internet Rights and Principles Dynamic Coalition, "10 Internet Rights & Principles", 2014,
<http://internetrightsandprinciples.org/site/wp-content/uploads/2014/06/IRPC_10RightsandPrinciples_28May2014-11.pdf>.
- [Kaye] Kaye, D., "The use of encryption and anonymity in digital communications", 2015,
<https://www.ohchr.org/EN/HRbodies/HRC/RegularSessions/Session29/Documents/A.HRC.29.32_AEV.doc>.

- [newegg] Mullin, J., "Newegg on trial: Mystery company TQP rewrites the history of encryption", 2013, <<http://arstechnica.com/tech-policy/2013/11/newegg-on-trial-mystery-company-tqp-re-writes-the-history-of-encryption/>>.
- [notewell] IETF, "Note Well", 2015, <<https://www.ietf.org/about/note-well.html>>.
- [patentpolicy] W3C, "W3C Patent Policy", 2004, <<https://www.w3.org/Consortium/Patent-Policy-20040205/>>.
- [Penney] Penney, J., "Chilling Effects: Online Surveillance and Wikipedia Use", 2016, <http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2769645>.
- [Pouwelse] Pouwelse, Ed, J., "Media without censorship", 2012, <<https://tools.ietf.org/html/draft-pouwelse-censorfree-scenarios>>.
- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <<https://www.rfc-editor.org/info/rfc793>>.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<https://www.rfc-editor.org/info/rfc1035>>.
- [RFC1958] Carpenter, B., Ed., "Architectural Principles of the Internet", RFC 1958, DOI 10.17487/RFC1958, June 1996, <<https://www.rfc-editor.org/info/rfc1958>>.
- [RFC1984] IAB and IESG, "IAB and IESG Statement on Cryptographic Technology and the Internet", BCP 200, RFC 1984, DOI 10.17487/RFC1984, August 1996, <<https://www.rfc-editor.org/info/rfc1984>>.
- [RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, DOI 10.17487/RFC2026, October 1996, <<https://www.rfc-editor.org/info/rfc2026>>.
- [RFC2277] Alvestrand, H., "IETF Policy on Character Sets and Languages", BCP 18, RFC 2277, DOI 10.17487/RFC2277, January 1998, <<https://www.rfc-editor.org/info/rfc2277>>.

- [RFC3365] Schiller, J., "Strong Security Requirements for Internet Engineering Task Force Standard Protocols", BCP 61, RFC 3365, DOI 10.17487/RFC3365, August 2002, <<https://www.rfc-editor.org/info/rfc3365>>.
- [RFC3724] Kempf, J., Ed., Austein, R., Ed., and IAB, "The Rise of the Middle and the Future of End-to-End: Reflections on the Evolution of the Internet Architecture", RFC 3724, DOI 10.17487/RFC3724, March 2004, <<https://www.rfc-editor.org/info/rfc3724>>.
- [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>>.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <<https://www.rfc-editor.org/info/rfc4033>>.
- [RFC4101] Rescorla, E. and IAB, "Writing Protocol Models", RFC 4101, DOI 10.17487/RFC4101, June 2005, <<https://www.rfc-editor.org/info/rfc4101>>.
- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <<https://www.rfc-editor.org/info/rfc4941>>.
- [RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <<https://www.rfc-editor.org/info/rfc4949>>.
- [RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, DOI 10.17487/RFC5321, October 2008, <<https://www.rfc-editor.org/info/rfc5321>>.
- [RFC5646] Phillips, A., Ed. and M. Davis, Ed., "Tags for Identifying Languages", BCP 47, RFC 5646, DOI 10.17487/RFC5646, September 2009, <<https://www.rfc-editor.org/info/rfc5646>>.
- [RFC6108] Chung, C., Kasyanov, A., Livingood, J., Mody, N., and B. Van Lieu, "Comcast's Web Notification System Design", RFC 6108, DOI 10.17487/RFC6108, February 2011, <<https://www.rfc-editor.org/info/rfc6108>>.

- [RFC6235] Boschi, E. and B. Trammell, "IP Flow Anonymization Support", RFC 6235, DOI 10.17487/RFC6235, May 2011, <<https://www.rfc-editor.org/info/rfc6235>>.
- [RFC6365] Hoffman, P. and J. Klensin, "Terminology Used in Internationalization in the IETF", BCP 166, RFC 6365, DOI 10.17487/RFC6365, September 2011, <<https://www.rfc-editor.org/info/rfc6365>>.
- [RFC6701] Farrel, A. and P. Resnick, "Sanctions Available for Application to Violators of IETF IPR Policy", RFC 6701, DOI 10.17487/RFC6701, August 2012, <<https://www.rfc-editor.org/info/rfc6701>>.
- [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>>.
- [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, <<https://www.rfc-editor.org/info/rfc7624>>.
- [RFC7725] Bray, T., "An HTTP Status Code to Report Legal Obstacles", RFC 7725, DOI 10.17487/RFC7725, February 2016, <<https://www.rfc-editor.org/info/rfc7725>>.
- [RFC7754] Barnes, R., Cooper, A., Kolkman, O., Thaler, D., and E. Nordmark, "Technical Considerations for Internet Service Blocking and Filtering", RFC 7754, DOI 10.17487/RFC7754, March 2016, <<https://www.rfc-editor.org/info/rfc7754>>.
- [RFC7844] Huitema, C., Mrugalski, T., and S. Krishnan, "Anonymity Profiles for DHCP Clients", RFC 7844, DOI 10.17487/RFC7844, May 2016, <<https://www.rfc-editor.org/info/rfc7844>>.

- [RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <<https://www.rfc-editor.org/info/rfc7858>>.
- [RFC8179] Bradner, S. and J. Contreras, "Intellectual Property Rights in IETF Technology", BCP 79, RFC 8179, DOI 10.17487/RFC8179, May 2017, <<https://www.rfc-editor.org/info/rfc8179>>.
- [RFC8280] ten Oever, N. and C. Cath, "Research into Human Rights Protocol Considerations", RFC 8280, DOI 10.17487/RFC8280, October 2017, <<https://www.rfc-editor.org/info/rfc8280>>.
- [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>>.
- [RFC8980] Arkko, J. and T. Hardie, "Report from the IAB Workshop on Design Expectations vs. Deployment Reality in Protocol Development", RFC 8980, DOI 10.17487/RFC8980, February 2021, <<https://www.rfc-editor.org/info/rfc8980>>.
- [RFC9071] Hellström, G., "RTP-Mixer Formatting of Multiparty Real-Time Text", RFC 9071, DOI 10.17487/RFC9071, July 2021, <<https://www.rfc-editor.org/info/rfc9071>>.
- [Saltzer] Saltzer, J.H., Reed, D.P., and D.D. Clark, "End-to-End Arguments in System Design", ACM TOCS, Vol 2, Number 4, November 1984, pp 277-288. , 1984.
- [UDHR] United Nations General Assembly, "The Universal Declaration of Human Rights", 1948, <<http://www.un.org/en/documents/udhr/>>.
- [UNGP] United Nations, "United Nations Guiding Principles on Business and Human Rights", 2011, <https://www.ohchr.org/documents/publications/guidingprinciplesbusinessshr_en.pdf>.
- [UNHR] United Nations, "The Core International Human Rights Instruments and their monitoring bodies", 2011, <<https://www.ohchr.org/en/professionalinterest/pages/coreinstruments.aspx>>.
- [UNHRC2016] United Nations Human Rights Council, "UN Human Rights Council Resolution "The promotion, protection and

enjoyment of human rights on the Internet" (A/HRC/32/L.20)", 2016, <<https://documents-dds-ny.un.org/doc/UNDOC/LTD/G16/131/89/PDF/G1613189.pdf?OpenElement>>.

[W3CAccessibility]

W3C, "Accessibility", 2015,
<<https://www.w3.org/standards/webdesign/accessibility>>.

[W3Ci18nDef]

W3C, "Localization vs. Internationalization", 2010,
<<http://www.w3.org/International/questions/qa-il8n.en>>.

[Zittrain] Zittrain, J., "The Future of the Internet - And How to Stop It", Yale University Press , 2008,
<https://dash.harvard.edu/bitstream/handle/1/4455262/Zittrain_Future%20of%20the%20Internet.pdf?sequence=1>.

Authors' Addresses

Gurshabad Grover
Centre for Internet and Society
Email: gurshabad@cis-india.org

Niels ten Oever
University of Amsterdam
Email: mail@nielstenoever.net

SUIT
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2019

B. Moran
H. Tschofenig
Arm Limited
July 02, 2018

A CBOR-based Manifest Serialisation Format
draft-moran-suit-manifest-02

Abstract

This specification describes the serialization format of a manifest.

A manifest is a bundle of metadata about the firmware for an IoT device, where to find the firmware, the devices to which it applies, and cryptographic information protecting the manifest.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 3, 2019.

Copyright Notice

Copyright (c) 2018 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. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction	2
2. Conventions and Terminology	3
2.1. Manifest Serialization Format	3
3. IANA Considerations	5
4. Security Considerations	5
5. Mailing List Information	5
6. Acknowledgements	6
7. References	6
7.1. Normative References	6
7.2. URIs	7
Authors' Addresses	7

1. Introduction

A firmware update mechanism is an essential security feature for IoT devices to deal with vulnerabilities. While the transport of firmware images to the devices themselves is important there are already various techniques available. Equally important is the inclusion of meta-data about the conveyed firmware image (in the form of a manifest) and the use of end-to-end security protection to detect modifications and (optionally) to make reverse engineering more difficult. End-to-end security allows the author, who builds the firmware image, to be sure that no other party (including potential adversaries) is able to install firmware updates on IoT devices without adequate privileges. This authorization process is ensured by the use of dedicated credentials and authorization permissions installed on the IoT device.

This document is part of larger document set: the architecture document can be found in [I-D.ietf-suit-architecture] and the information model of the manifest is described in [I-D.ietf-suit-information-model]. This document focuses on the serialization format.

2. Conventions and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2.1. Manifest Serialization Format

The following CDDL fragment defines the manifest.

Wherever enumerations are used, they are started at 1. This allows detection of several common software errors that are caused by uninitialised variables.

The processing graph is a mechanism that maps from resources to installed firmware. On one side of the graph are the resources. These are the raw content that a device acquires. Resources can be remote (for example, on a server) or local (for example, an already installed firmware). On the other side of the graph are targets. These are the locations that firmware is installed to. In between these two sides are processors. These are the steps that a device takes to translate raw content into firmware that is installed. In the simplest case, this is a direct mapping; the resource is installed into the target directly. In an example complex case, a device must use decryption, decompression, and differential patching to create the final resource. Differential patching requires that the device refer to an already-installed firmware. In this graph, there are two resources, three processors, and one target. In some cases, one resource may be used by multiple processors, such as a compression table. The nodes of the graph are the resources before or after transformation by a processor and the edges of the graph are the processors themselves.

Resources, processors and targets are marked with node identifiers. Resources have an output node. Targets have an input node. Processors have both.

```
AuthenticatedManifest = [  
    authenticatedManifest: COSE_Mac / COSE_Sign,  
    text: bstr .cbor textMap  
]  
COSE_Mac = any  
COSE_Sign = any  
  
textKeys = (  
    uninitialised: 0 /  
    manifestDescription: 1 /
```

```

    payloadDescription: 2 /
    vendorName: 3 /
    modelName: 4 /
    payloadVersion: 5
)

textMap = { * textKeys / nint => tstr }

Manifest = [
    manifestVersion :    1,
    digestInfo :        DigestInfo,

    ; textReference is the digest of the associated
    ; text map in AuthenticatedManifest
    textReference :      bstr,
    nonce :              bstr,
    sequence :           SequenceNumber,
    preConditions :      [ * PreCondition ],
    postConditions :     [ * PostCondition ],
    directives :         [ * Directive ],
    resources :          [ * ResourceInfo ],
    processors :         [ * ProcessingStep ],
    targets :            [ * TargetInfo ],
    extensions :         { * int => bstr}
]

ResourceInfo = [
    type:                payload:1 / dependency:2 / key:3 / alias:4
    indicator:            UriList,          ; where to find the resource
    size:                uint / nil,        ; size of the resource
                                          ; (nil when alias)
    digest:              bstr,              ; digest of the resource
    onode                bstr              ; Node of the processing
                                          ; graph that the resource feeds
]

Processor = [
    decrypt: 1 / decompress: 2 / undiff: 3 /
    relocate: 4 / unrellocate: 5,
    parameters: bstr ; TBD: more detail needed
    inode: bstr,      ; Node of the processing graph
                    ; that this processor consumes
    onode: bstr       ; Node of the processing graph
                    ; that this processor feeds
]

Target = [
    componentIdentifier: [ * bstr],
    storageIdentifier:   tstr,      ; where to store the resource

```



```

        encoding:          bstr / nil,    ; the format of the resource
                                ; (nil when alias)
        inode:             bstr           ; Node of the processing graph
                                ; that this target consumes
    ]

    PreCondition    = IdCondition / TimeCondition /
                      ImageCondition / CustomCondition
    PostCondition   = ImageCondition / CustomCondition
    IdCondition     = [vendor: 1 / class: 2 / device: 3,
                      id:      Uuid]
    TimeCondition   = [installAfter: 4 / bestBefore: 5,
                      time:      Timestamp]
    ImageCondition  = [currentContent: 6 / notCurrentContent: 7,
                      digest:     bstr / nil, location: StorageIdentifier]
    CustomCondition = [nint, parameters: bstr]
    Directive       = [ int => bstr ]

    SequenceNumber  = uint
    Timestamp       = uint .size 8
    Uuid            = bstr .size 16
    StorageIdentifier = bstr
    ComponentIdentifier = bstr
    UriList         = { + int => tstr }
    DigestInfo      = [
        digestAlgorithm : uint,
        ? digestParameters : bstr
    ]

```

3. IANA Considerations

TBD: Several registries will be required for: * Standard Conditions * Standard Directives * Standard Processors * Standard text values

4. Security Considerations

This document is about a manifest format describing and protecting firmware images and as such it is part of a larger solution for offering a standardized way of delivering firmware updates to IoT devices. A more detailed discussion about security can be found in the architecture document [I-D.ietf-suit-architecture] and in the information model document [I-D.ietf-suit-information-model].

5. Mailing List Information

The discussion list for this document is located at the e-mail address suit@ietf.org [1]. Information on the group and information

on how to subscribe to the list is at
<https://www1.ietf.org/mailman/listinfo/suit>

Archives of the list can be found at: <https://www.ietf.org/mail-archive/web/suit/current/index.html>

6. Acknowledgements

We would like the following persons for their support in designing this mechanism

- Geraint Luff
- Amyas Phillips
- Dan Ros
- Carsten Bormann
- David Brown
- Markus Gueller
- Frank Audun Kvamtro
- Oyvind Ronningstad

7. References

7.1. Normative References

- [I-D.ietf-suit-architecture]
Moran, B., Meriac, M., Tschofenig, H., and D. Brown, "A Firmware Update Architecture for Internet of Things Devices", draft-ietf-suit-architecture-01 (work in progress), July 2018.
- [I-D.ietf-suit-information-model]
Moran, B., Tschofenig, H., Birkholz, H., and J. Jimenez, "Firmware Updates for Internet of Things Devices - An Information Model for Manifests", draft-ietf-suit-information-model-00 (work in progress), June 2018.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

7.2. URIs

[1] <mailto:suit@ietf.org>

Authors' Addresses

Brendan Moran
Arm Limited

EMail: Brendan.Moran@arm.com

Hannes Tschofenig
Arm Limited

EMail: hannes.tschofenig@gmx.net

Human Rights Protocol Considerations Research Group
Internet-Draft

Intended status: Informational

Expires: November 30, 2018

N. ten Oever
University of Amsterdam

G. Perez de Acha
Derechos Digitales
May 29, 2018

Freedom of Association on the Internet
draft-tenoever-hrpc-association-05

Abstract

This document scopes the relation between Internet protocols and the right to freedom of assembly and association. Increasingly, the Internet mediates our lives, our relationships and our ability to exercise our human rights. The Internet provides a global public space, but one that is built predominantly on private infrastructure. Since Internet protocols play a central role in the management, development and use of the Internet, the relation between protocols and the aforementioned rights should be documented and any adverse impacts of this relation should be mitigated.

Status of This Memo

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

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

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

This Internet-Draft will expire on November 30, 2018.

Copyright Notice

Copyright (c) 2018 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 (<https://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. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Vocabulary used	3
3. Research questions	5
4. Methodology	5
5. Literature Review	5
6. Cases and examples	7
6.1. Conversing	7
6.1.1. Mailing Lists	7
6.1.2. Multi-party video conferencing	8
6.1.3. Internet Relay Chat	8
6.2. Peer-to-peer networks and systems	9
6.2.1. Peer-to-peer system architectures	9
6.2.2. Version control	11
6.3. Grouping together (identities)	11
6.3.1. DNS	12
6.3.2. Autonomous Systems	12
7. Discussion: Protocols vs Platforms	13
8. Conclusions	14
9. Acknowledgements	15
10. Security Considerations	15
11. IANA Considerations	15
12. Research Group Information	15
13. References	15
13.1. Informative References	15
13.2. URIs	22
Authors' Addresses	22

1. Introduction

"We shape our tools and, thereafter, our tools shape us." 
- John Culkin (1967)

The Internet is a technology which shapes modern information societies. The ordering that the Internet provides is socio-technical, in other words, the Internet infrastructure and architecture consists of social and technological arrangements [StarRuhleder]. This ordering is not always apparent because infrastructure also tends to hide itself in the societal woodwork [Mosco], or with [Weiser]: 'The most profound technologies are those that disappear'. Next to that infrastructure is often taken for

granted by those using it. Infrastructure therefore is mostly known by an epistemic community of experts [Haas] and only get recognized by the larger public when it fails. With the increasing societal use of the Internet the importance of the Internet is growing, and the decisions made about its infrastructure and architecture therefore also become more important. [RFC8280] established the relationship between human rights and Internet protocols, and in this document we seek to uncover the relation between two specific human rights and the Internet infrastructure and architecture.

The rights to freedom of assembly and association protect collective expression, in turn, systems and protocols that enable communal communication between people and servers allow these rights to prosper. The Internet itself was originally designed as "a medium of communication for machines that share resources with each other as equals" [NelsonHedlun], the Internet thus forms a basic infrastructure for the right freedom of assembly and association.

The manner in which communication is designed and implemented impacts the ways in which rights can be exercised. For instance a decentralized and resilient architecture that protects anonymity and privacy, offers a strong protection for the exercise of such freedoms in the online environment. At the same time, centralized solutions have enabled people to group together in recognizable places and helped the visibility of groups. In other words, different architectural designs come with different affordances, or characteristics. These characteristics should be taken into account at the time of design, and when designing, updating and maintaining other parts of the architecture and infrastructure.

This draft continues the work started in [RFC8280] by investigating the exact impact of Internet protocols on specific human rights, namely the right to freedom of assembly and association given their importance for the Internet, in order to mitigate (potential) negative impacts.

2. Vocabulary used

Architecture The design of a structure

Autonomous System (AS) Autonomous Systems are the unit of routing policy in the modern world of exterior routing [RFC1930].

Within the Internet, an autonomous system (AS) is a collection of connected Internet Protocol (IP) routing prefixes under the control of one or more network operators on behalf of a single administrative entity or domain that presents a common, clearly defined routing policy to the Internet [RFC1930].

The classic definition of an Autonomous System is a set of routers under a single technical administration, using an interior gateway protocol and common metrics to route packets within the AS, and using an exterior gateway protocol to route packets to other ASs [RFC1771].

Border Gateway Protocol (BGP) An inter-Autonomous System routing protocol [RFC4271].

Connectivity The extent to which a device or network is able to reach other devices or networks to exchange data. The Internet is the tool for providing global connectivity [RFC1958]. Different types of connectivity are further specified in [RFC4084]. The combination of the end-to-end principle, interoperability, distributed architecture, resilience, reliability and robustness are the enabling factors that result in connectivity to and on the Internet.

Decentralization Implementation or deployment of standards, protocols or systems without one single point of control.

Distributed system A system with multiple components that have their behavior co-ordinated via message passing. These components are usually spatially separated and communicate using a network, and may be managed by a single root of trust or authority. [Troncosoetal]

Infrastructure Underlying basis or structure for a functioning society, organization or community. Because infrastructure is a precondition for other activities it has a procedural, rather than static, nature due to its social and cultural embeddedness [PipekWulf] [Bloketal]. This means that infrastructure is always relational: infrastructure always develops in relation to something or someone [Bowker].

Internet The Network of networks, that consists of Autonomous Systems that are connected through the Internet Protocol (IP).

A persistent socio-technical system over which services are delivered [Mainwaringetal],

A techno-social assemblage of devices, users, sensors, networks, routers, governance, administrators, operators and protocols

An emergent-process-driven thing that is born from the collections of the ASes that happen to be gathered together at any given time. The fact that they tend to interact at any given time means it is

an emergent property that happens because they use the protocols defined at IETF.

3. Research questions

1. How does the internet architecture enable and/or inhibit freedom of association and assembly?
2. If the Internet is used to exercise the right to freedom of association, what are the implications for its architecture and infrastructure?

4. Methodology

In order to answer the research questions, first a number of cases have been collected to analyze where Internet infrastructure and protocols have either enabled or inhibited groups of people to collaborate, cooperate or communicate. This overview does not aim to cover all possible ways in which people can collectively organize or reach out to each other using Internet infrastructure and Internet protocols, but rather cover typical uses in an attempt at an ethnography of infrastructure [Star]. Subsequently we analyze the cases with the theoretical framework provided in the literature review and provide recommendations based on the findings.

5. Literature Review

The rights to freedom of assembly and association protects and enables collective action and expression [UDHR] [ICCPR]. These rights ensure everyone in a society has the opportunity to express the opinions they hold in common with others, which in turn facilitates dialogue among citizens, as well as with political leaders or governments [OSCE]. This is relevant because in the process of democratic deliberation, causes and opinions are more widely heard when a group of people come together behind the same cause or issue [Tocqueville].

In international law, the rights to freedom of assembly and association protect any collective, gathered either permanently or temporarily for "peaceful" purposes. It is important to underline the property of "freedom" because the right to freedom of association and assembly are voluntary and uncoerced: anyone can join or leave a group of choice, which in turn means one should not be forced to either join, stay or leave.

The difference between freedom of assembly and freedom of association is merely gradual one: the former tends to have an informal and ephemeral nature, whereas the latter refers to established and

permanent bodies with specific objectives. Nonetheless, one and the other are protected to the same degree.

An assembly is an intentional and temporary gathering of a collective in a private or public space for a specific purpose: demonstrations, indoor meetings, strikes, processions, rallies or even sits-in [UNHRC]. Association on the other hand has a more formal and established nature. It refers to a group of individuals or legal entities brought together in order to collectively act, express, pursue or defend a field of common interests [UNGA]. Within this category we can think about civil society organizations, clubs, cooperatives, NGOs, religious associations, political parties, trade unions or foundations.

The right to freedom of assembly and association is quintessential for the Internet, even if privacy and freedom of expression are the most discussed human rights when it comes to the online world. Online association and assembly are crucial to mobilise groups and people where physical gatherings have been impossible or dangerous [APC]. Throughout the world -from the Arab Spring to Latin American student movements and the #WomensMarch- the Internet has also played a crucial role by providing a means for the fast dissemination of information that was otherwise mediated by broadcast media, or even forbidden by the government [Pensado]. According to Hussain and Howard the Internet helped to "build solidarity networks and identification of collective identities and goals, extend the range of local coverage to international broadcast networks" and as platform for contestation for "the future of civil society and information infrastructure" [HussainHoward].

The IETF itself, defined as a 'open global community' of network designers, operators, vendors, and researchers, is also protected by freedom of assembly and association [RFC3233]. Discussions, comments and consensus around RFCs are possible because of the collective expression that freedom of association and assembly allow. The very word "protocol" found its way into the language of computer networking based on the need for collective agreement among network users [HafnerandLyon].

We are aware that some of these examples go beyond the use of Internet protocols and flow over into the applications layer or examples in the offline world whereas the purpose of the following document is to break down the relationship between Internet protocols and the right to freedom of assembly and association. Nonetheless, given that protocols are a part of the socio-technical ordering of reality, we do recognize that in some cases the line between them and applications, implementations, policies and offline realities are often blurred and hard (if not impossible) to differentiate.

6. Cases and examples

The Internet has become a central mediator for collective action and collaboration. This means the Internet has become a strong enabler of the rights to freedom of association and assembly.

Here we will discuss different cases to give an overview of how the Internet protocol and architecture facilitates the freedom of assembly and association.

6.1. Conversing

An interactive conversation between two or more people forms the basis for people to organize and associate. According to Anderson "the relationship between political conversation and engagement in the democratic process is strong." [Anderson]. By this definition, what defines the "political" is essentially assembly or association: a basis for the development of social cohesion in society.

6.1.1. Mailing Lists

Since the beginning of the Internet mailing lists have been a key site of assembly and association [RFC0155] [RFC1211]. In fact, mailing lists were one of the Internet's first functionalities [HafnerandLyon].

In 1971, four years after the invention of email, the first mailing list was created to talk about the idea of using Arpanet for discussion. What had initially propelled the Arpanet project forward as a resource sharing platform was gradually replaced by the idea of a network as a means of bringing people together [Abbate]. More than 45 years after, mailing lists are pervasive and help communities to engage, have discussion, share information, ask questions, and build ties. Even as social media and discussion forums grow, mailing lists continue to be widely used [AckermannKargerZhang]. They are a crucial tool to organise groups and individuals around themes and causes [APC].

Mailinglist are still in wide use, also in the IETF because they allow for easy association and allow people to subscribe (join) and unsubscribe (leave) as they please. They also allow for association of specific groups on closed lists. Finally the archival function allows for accountability. The downsides of mailinglists are similar to the ones generally associated with e-mail, except that end-to-end encryption such as OpenPGP [RFC4880] and S/MIME [RFC5751] is not possible because the final recipients are not known. There have been experimental solutions to address this issue such as Schleuder [Schleuder], but this has not been standardized or widely deployed.

6.1.2. Multi-party video conferencing

Multi-party video conferencing protocols such as WebRTC [RFC6176] [RFC7118] allow for robust, bandwidth-adaptive, wideband and super-wideband video and audio discussions in groups. 'The WebRTC protocol was designed to enable responsive real-time communications over the Internet, and is instrumental in allowing streaming video and conferencing applications to run in the browser. In order to easily facilitate direct connections between computers (bypassing the need for a central server to act as a gatekeeper), WebRTC provides functionality to automatically collect the local and public IP addresses of Internet users (ICE or STUN). These functions do not require consent from the user, and can be instantiated by sites that a user visits without their awareness. The potential privacy implications of this aspect of WebRTC are well documented, and certain browsers have provided options to limit its behavior.' [AndersonGuarnieri].

While facilitating freedom of assembly and association multi-party video conferencing tools might pose concrete risks for those who use them. On the one hand WebRTC is providing resilient channels of communications, but on the other hand it also exposes information about those who are using the tool which might lead to increased surveillance, identification and the consequences that might be derived from that. This is especially concerning because the usage of a VPN does not protect against the exposure of IP addresses [Crawford].

The risk of surveillance is also true in an offline space, but this is generally easy to analyze for the end-user. Security and privacy expectations of the end-user could be made more clear to the user (or improved) which would result in a more secure and/or private exercise of the right to freedom of assembly or association.

6.1.3. Internet Relay Chat

Internet Relay Chat (IRC) is an application layer protocol that enables communication in the form of text through a client/server networking model [RFC2810]. In other words, a chat service. IRC clients are computer programs that a user can install on their system. These clients communicate with chat servers to transfer messages to other clients.

For order to be kept within the IRC network, special classes of users become "operators" and are allowed to perform general maintenance functions on the network: basic network tasks such as disconnecting (temporary or permanently) and reconnecting servers as needed [RFC2812]. One of the most controversial power of operators is the

ability to remove a user from the connected network by 'force', i.e., operators are able to close the connection between any client and server [RFC2812].

IRC servers may deploy different policies for the ability of users to create their own channels or 'rooms', and for the delegation of 'operator'-rights in such a room. Some IRC servers support SSL/TLS connections for security purposes [RFC7194]. This helps stop the use of packet sniffer programs to obtain the passwords of IRC users, but has little use beyond this scope due to the public nature of IRC channels. TLS connections require both client and server support (that may require the user to install TLS binaries and IRC client specific patches or modules on their computers). Some networks also use TLS for server to server connections, and provide a special channel flag (such as +S) to only allow TLS-connected users on the channel, while disallowing operator identification in clear text, to better utilize the advantages that TLS provides.

6.2. Peer-to-peer networks and systems

At the organizational level, peer production is one of the most relevant innovations from Internet mediated social practices. According to [Benkler], it implies 'open collaborative innovation and creation, performed by diverse, decentralized groups organized principally by neither price signals nor organizational hierarchy, harnessing heterogeneous motivations, and governed and managed based on principles other than the residual authority of ownership implemented through contract.' [Benkler].

In his book *The Wealth of Networks*, Benkler significantly expands on his definition of commons-based peer production. According to Benkler, what distinguishes commons-based production is that it doesn't rely upon or propagate proprietary knowledge: "The inputs and outputs of the process are shared, freely or conditionally, in an institutional form that leaves them equally available for all to use as they choose at their individual discretion." [Benkler] To ensure that the knowledge generated is available for free use, commons-based projects are often shared under an open license.

6.2.1. Peer-to-peer system architectures

Peer-to-peer (P2P) is essentially a model of how people interact in real life because "we deal directly with one another whenever we wish to" [Vu]. Usually if we need something we ask our peers, who in turn refer us to other peers. In this sense, the ideal definition of P2P is that "nodes are able to directly exchange resources and services between themselves without the need for centralized servers" and where each participating node typically acts both as a server and as

a client [Vu]. In RFC 5694 P2P has been defined as peers or nodes that should be able to communicate directly between themselves without passing intermediaries, and that the system should be self-organizing and have decentralized control [RFC5694]. With this in mind, the ultimate model of P2P is a completely decentralized system, which is more resistant to speech regulation, immune to single points of failure and have a higher performance and scalability. Nonetheless, in practice some P2P systems are supported by centralized servers and some others have hybrid models where nodes are organized into two layers: the upper tier servers and the lower tier common nodes [Vu].

Since the ARPANET project, the original idea behind the Internet was conceived as what we would now call a peer-to-peer system [RFC0001]. Over time it has increasingly shifted towards a client/server model with "millions of consumer clients communicating with a relatively privileged set of servers" [NelsonHedlun].

Whether for resource sharing or data sharing, P2P systems are enabling freedom of assembly and association. Not only do they allow for effective dissemination of information, but because they leverage computing resources by diminishing costs allowing for the formation of open collectives at the network level. At the same time, in completely decentralized systems the nodes are autonomous and can join or leave the network as they want, which also makes the system unpredictable: a resource might be only sometimes available, and some other resources might be missing or incomplete [Vu]. Lack of information might in turn make association or assembly more difficult.

Additionally, when one architecturally assesses the role of P2P systems one can say that: "The main advantage of centralized P2P systems is that they are able to provide a quick and reliable resource locating. Their limitation, however, is that the scalability of the systems is affected by the use of servers. While decentralized P2P systems are better than centralized P2P systems in this aspect, they require a longer time in resource locating. As a result, hybrid P2P systems have been introduced to take advantage of both centralized and decentralized architectures. Basically, to maintain the scalability, similar to decentralized P2P systems, there are no servers in hybrid P2P systems. However, peer nodes that are more powerful than others can be selected to act as servers to serve others. These nodes are often called super peers. In this way, resource locating can be done by both decentralized search techniques and centralized search techniques (asking super peers), and hence the systems benefit from the search techniques of centralized P2P systems." [Vu]

6.2.2. Version control

Ever since developers needed to collaboratively write, maintain and discuss large code basis for the Internet there have been different approaches of doing so. One approach is discussing code through mailing lists, but this has proven to be hard in case of maintaining the most recent versions. There are many different versions and characteristics of version control systems.

A version control system is a piece of software that enables developers on a software team to work together and also archive a complete history of their work [Sink]. This allows teams to be working simultaneously on updated versions. According to Sink, broadly speaking, the history of version control tools can be divided into three generations. In the first one, concurrent development meant that only one person could be working on a file at a time. The second generation tools permit simultaneous modifications as long as users merge the current revisions into their work before they are allowed to commit. The third generation tools allow merge and commit to be separated [Sink].

Interestingly no version control system has ever been standardized in the IETF whereas the version control systems like Subversion and Git are widely used within the community, as well as by working groups. There has been a spirited discussion on whether working groups should use centralized forms of the Git protocol, such as those offered by Gitlab or Github. Proponents argue that this simplifies the workflow and allows for a more transparent workflow. Opponents argue that the reliance on a centralized service which is not merely using the Git protocol, but also uses non-standardized options like an Issue-Tracker, makes the process less transparent and reliant on a third party.

The IETF has not made a decision on the use of centralized instances of Git, such as Github or Gitlab. There have been two efforts to standardize the workflow vis a vis these third party services, but these haven't come to fruition: [Wugh] [GithubIETF].

6.3. Grouping together (identities)

Collective identities are also protected by freedom of association and assembly. According to Melucci these are 'shared definitions produced by several interacting individuals who are concerned with the orientation of their action as well as the field of opportunities and constraints in which their action takes place.' [Melucci] In this sense, assemblies and associations are an important base in the maintenance and development of culture, as well as preservation of minority identities [OSCE].

6.3.1. DNS

Domain names allow hosts to be identified by human parsable information. Whereas an IP address might not be the expression of an identity, a domain name can be, and often is. On the other hand the grouping of a certain identity under a specific domain or even a Top Level Domain brings about risks because connecting an identity to a hierarchically structured identifier systems creates a central attack surface. Some of these risks are the surveillance of the services running on the domain, domain based censorship [RFC7754], or impersonation of the domain through DNS cache poisoning. Several technologies have been developed in the IETF to mitigated these risks such as DNS over TLS [RFC7858], DNSSEC [RFC4033], and TLS [RFC5246]. These mitigations would, when implemented, not make censorship impossible, but rather make it visible. The use of a centralized authority always makes censorship through a registry or registrar possible, as well as by using a fake resolver or using proposed standards such as DNS Response Policy Zones [RPZ].

The structuring of DNS as a hierarchical authority structure also brings about a specific characteristic, namely the possibility of centralized policy making vis a vis the management and operation of Top Level Domains, which is what (in part) happens at ICANN. The impact of ICANN processes on human rights will not be discussed here.

6.3.2. Autonomous Systems

In order for edge-users to connect to the Internet, they need to be connected to an Automous System (AS) which, in turn, has peering or transit relations with other AS'es. This means that in the process of accessing the Internet, edge-users need to accept the policies and practices of the intermediary that provides them access to the other networks. In other words, for users to be able to join the 'network of networks', they always need to connect through an intermediary.

While accessing the Internet through an intermediary, the user is forced to accept the policies, practices and principles of a network. This could impede the rights of the edge-user, depending on the implemented policies and practices on the network and how (if at all) they are communicated to them. For example: filtering, blocking, extensive logging, slowing down connection or specific services, or other invasive practices that are not clearly communicated to the user.

In some cases it also means that there is no other way for the edge-user to connect to the network of networks, and is thus forced into accepting the policies of a specific network, because it is not trivial for an edge-user to operate an AS and engage in peering

relation with other ASes. This design, combined with the increased importance of the Internet to make use of basic services, forces edge-user to engage in association with a specific network eventhough the user does not consent to the policies of the network.

It can be noted also that there is no standard and deployed way for the edge-user to choose the routes her packets will go through. [RFC0791], section 3.1, standardized "source routing" but it was never deployed, mostly because of serious security issues. There is not even a way for the edge-user to know about the routes that packets have actually taken, and which ASes a packet has traversed. [RFC0791], section 3.1, standardized "record route" but it was never deployed. In practice, the user must accept policies of ASes he has no relationship with, and didn't choose. For instance, there is no way to direct the packets to avoid the Five Eyes, not even to know after the fact where the packet went. [FiveEyes] [SchengenRouting] (Traceroutes give you an idea but the path may change before and after the traceroute.)

7. Discussion: Protocols vs Platforms

The Internet is increasingly becoming a vehicle for commercial, proprietary, non-interoperable platforms. The Internet has always allowed for closed-off networks, but the current trend show the rise of a small number of very large non-interoperable platforms. Chat has moved from XMPP and IRC to Facebook Messenger, Whatsapp and WeChat and there has been a strong rise of social media networks with large numbers of users, such as Facebook, Twitter and Instagram. A similar trend can be found among e-mail providers, with the significant difference that e-mail is interoperable.

Often these non-interoperable platforms are built on open-protocols but do not allow for inter-operability or data-portability. In the case of these large platforms this leads to strong network externalities, also know as a network effect; because the users are there, users will be there. The use of social-media platforms has enabled groups to associate, but is has also led to a 'tactical freeze' because of the inability to change the platforms [Tufekci]. Whereas these networks are a ready-to-hand networked public sphere, they do not allow their inhabitants to change, or fully understand, their workings.

This potentially has a significant impact on the distributed nature of the Internet [RFC1287].

8. Conclusions

This document scopes the relation between Internet protocols and the right to freedom of assembly and association. For this reason, the current research started out with two main questions. First, how does the internet architecture enable and/or inhibit freedom of association and assembly? And secondly: if the Internet is used to exercise the right to freedom of association, what are the implications for its architecture and infrastructure?

Communities, collaboration and joint action lie at the heart of the Internet. Even at a linguistic level, the words "networks" and "associations" are close synonyms. Both interconnected groups and assemblies of people depend on "links" and "relationships" [Swire]. Taking legal definitions given in international human rights law jurisprudence, we could assert that the right to freedom of assembly and association protect collective expression. These rights protect any collective, gathered either permanently or temporarily for "peaceful" purposes. It is voluntary and uncoerced.

Regarding the first question, we argued that given that the Internet itself was originally designed as a medium of communication for machines that share resources with each other as equals, the Internet is one of the most basic infrastructures for the right to freedom of assembly and association. Since Internet protocols play a central role in the management, development and use of the Internet, we established the relation between some protocols and the right to freedom of assembly and association.

Regarding the second question, after reviewing protocols that allow mailing lists, to multi-party video conferencing, IRC, peer-to-peer architectures, version control or the functioning of autonomous systems, we can conclude that the way in which infrastructure is designed and implemented impacts the exercise of freedom of assembly and association. This is because different architectural designs come with different affordances, or characteristics. If a decentralized architecture protects anonymity and privacy, both freedoms in the online environment will be enabled. On the other hand, centralized solutions have allowed users to group together and visible groups. enabled people to group together in recognizable places and helped the visibility of groups.

Lastly, the increasing shift towards closed and non-interoperable platforms in chat and social media networks have a significant impact on the distributed and open nature of the Internet. Often these non-interoperable platforms are built on open-protocols but do not allow for inter-operability or data-portability. The use of social-media platforms has enabled groups to associate, but it has also rendered

users unable to change platforms, therefore leading to a sort of "forced association" that stirs faraway from freedom.

9. Acknowledgements

- Fred Baker, Jefsey, and Andrew Sullivan for work on Internet definitions
- Stephane Bortzmeyer for several concrete text suggestions that found their way in this document (such as the AS filtering example)
- Mark Perkins for finding a lot of typos
- the hrpc mailinglist at large for a very constructive discussion on a hard topic.

10. Security Considerations

As this draft concerns a research document, there are no security considerations.

11. IANA Considerations

This document has no actions for IANA.

12. Research Group Information

The discussion list for the IRTF Human Rights Protocol Considerations Research Group is located at the e-mail address hrpc@ietf.org [1]. Information on the group and information on how to subscribe to the list is at <https://www.irtf.org/mailman/listinfo/hrpc> [2]

Archives of the list can be found at: <https://www.irtf.org/mail-archive/web/hrpc/current/index.html> [3]

13. References

13.1. Informative References

- [Abbate] Janet Abbate, ., "Inventing the Internet", Cambridge: MIT Press (2013): 11. , 2013,
<<https://mitpress.mit.edu/books/inventing-internet>>.

[AckermannKargerZhang]

Ackerman, M., Karger, D., and A. Zhang, "Mailing Lists: Why Are They Still Here, What's Wrong With Them, and How Can We Fix Them?", Mit. edu (2017): 1. , 2017, <<https://people.csail.mit.edu/axz/papers/maillinglists.pdf>>.

[Anderson]

Andersson, E., "The political voice of young citizens Educational conditions for political conversation - school and social media", Utbildning & Demokrati: Tidskrift foer Didaktik och Utbildningspolitik, Volume 21, Number 1, 2012, pp. 97-119(23) , 2012, <<http://www.ingentaconnect.com/content/doi/11026472/2012/00000021/00000001/art00006>>.

[AndersonGuarnieri]

Anderson, C. and C. Guarnieri, "Fictitious Profiles and WebRTC's Privacy Leaks Used to Identify Iranian Activists", 2016, <<https://iranthreats.github.io/resources/webrtc-deanonymization/>>.

[APC]

Association for Progressive Communications and . Gayathry Venkiteswaran, "Freedom of assembly and association online in India, Malaysia and Pakistan. Trends, challenges and recommendations.", 2016, <https://www.apc.org/es/system/files/FOAA_online_IndiaMalaysiaPakistan.pdf>.

[Benkler]

Benkler, Y., "Peer Production and Cooperation", 2009, <<http://www.benkler.org/Peer%20production%20and%20cooperation%2009.pdf>>.

[Bloketal]

Blok, A., Nakazora, M., and B. Winthereik, "Infrastructuring Environments", Science as Culture 25:1, 1-22. , 2016.

[Bowker]

Bowker, G., "Information mythology and infrastructure", In: L. Bud (Ed.), Information Acumen: The Understanding and use of Knowledge in Modern Business, Routledge, London, 1994, pp. 231-247 , 1994.

[Crawford]

Crawford, D., "The WebRTC VPN "Bug" and How to Fix", 2015, <<https://www.bestvpn.com/the-webrtc-vpn-bug-and-how-to-fix-it/>>.

- [FiveEyes] Wikipedia, ., "Five Eyes", 2018, <https://en.wikipedia.org/wiki/Five_Eyes>.
- [GithubIETF] Thomson, M. and A. Atlas, "Using GitHub at the IETF", 2017.
- [Haas] Haas, P., "Introduction: epistemic communities and international policy coordination", International Organization, special issue: Knowledge, Power, and International Policy Coordination, Cambridge Journals. 46 (1): 1-35. , 1992.
- [HafnerandLyon] Hafnerand, K. and M. Lyon, "Where Wizards Stay Up Late. The Origins of the Internet", First Touchstone Edition (1998): 93. , 1998, <<https://doi.org/10.1111/misr.12020>>.
- [HussainHoward] Hussain, M. and P. Howard, "What Best Explains Successful Protest Cascades? ICTs and the Fuzzy Causes of the Arab Spring", Int Stud Rev (2013) 15 (1): 48-66. , 2013, <<https://doi.org/10.1111/misr.12020>>.
- [ICCPR] United Nations General Assembly, "International Covenant on Civil and Political Rights", 1966, <<http://www.ohchr.org/EN/ProfessionalInterest/Pages/CCPR.aspx>>.
- [Mainwaringetal] Mainwaring, S., Chang, M., and K. Anderson, "Infrastructures and Their Discontents: Implications for Ubicomp", DBLP Conference: Conference: UbiComp 2004: Ubiquitous Computing: 6th International Conference, Nottingham, UK, September 7-10, 2004. Proceedings , 2004, <<http://www.dourish.com/classes/readings/Mainwaring-Infrastructure.pdf>>.
- [Melucci] Melucci, A., "The Process of Collective Identity", Temple University Press, Philadelphia , 1995.
- [Mosco] Mosco, V., "The Digital Sublime: Myth, Power, and Cyberspace", 2005, <<https://mitpress.mit.edu/books/digital-sublime>>.

[NelsonHedlun]

Minar, N. and M. Hedlun, "A Network of Peers: Models Through the History of the Internet", Peer to Peer: Harnessing the Power of Disruptive Technologies, ed: Andy Oram , 2001, <http://library.uniteddiversity.coop/REconomy_Resource_Pack/More_Inspirational_Videos_and_Useful_Info/Peer_to_Peer-Harnessing_the_Power_of_Disruptive_Technologies.pdf>.

[OSCE]

OSCE Office for Democratic Institutions and Human Rights, "Guidelines on Freedom of Peaceful Assembly", page 24 , 2010, <<https://www.osce.org/odihr/73405?download=true>>.

[Pensado]

Jaime Pensado, ., "Student Activism. Utopian Dreams.", ReVista. Harvard Review of Latin America (2012). , 2012, <<http://revista.drclas.harvard.edu/book/student-activism>>.

[PipekWulf]

Pipek, V. and W. Wolf, "Infrastructuring: Towards an Integrated Perspective on the Design and Use of Information Technology", Journal of the Association for Information Systems (10) 5, pp. 306-332 , 2009.

[RFC0001]

Crocker, S., "Host Software", RFC 1, DOI 10.17487/RFC0001, April 1969, <<https://www.rfc-editor.org/info/rfc1>>.

[RFC0155]

North, J., "ARPA Network mailing lists", RFC 155, DOI 10.17487/RFC0155, May 1971, <<https://www.rfc-editor.org/info/rfc155>>.

[RFC0791]

Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/info/rfc791>>.

[RFC1211]

Westine, A. and J. Postel, "Problems with the maintenance of large mailing lists", RFC 1211, DOI 10.17487/RFC1211, March 1991, <<https://www.rfc-editor.org/info/rfc1211>>.

[RFC1287]

Clark, D., Chapin, L., Cerf, V., Braden, R., and R. Hobby, "Towards the Future Internet Architecture", RFC 1287, DOI 10.17487/RFC1287, December 1991, <<https://www.rfc-editor.org/info/rfc1287>>.

[RFC1771]

Rekhter, Y. and T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, DOI 10.17487/RFC1771, March 1995, <<https://www.rfc-editor.org/info/rfc1771>>.

- [RFC1930] Hawkinson, J. and T. Bates, "Guidelines for creation, selection, and registration of an Autonomous System (AS)", BCP 6, RFC 1930, DOI 10.17487/RFC1930, March 1996, <<https://www.rfc-editor.org/info/rfc1930>>.
- [RFC1958] Carpenter, B., Ed., "Architectural Principles of the Internet", RFC 1958, DOI 10.17487/RFC1958, June 1996, <<https://www.rfc-editor.org/info/rfc1958>>.
- [RFC2810] Kalt, C., "Internet Relay Chat: Architecture", RFC 2810, DOI 10.17487/RFC2810, April 2000, <<https://www.rfc-editor.org/info/rfc2810>>.
- [RFC2812] Kalt, C., "Internet Relay Chat: Client Protocol", RFC 2812, DOI 10.17487/RFC2812, April 2000, <<https://www.rfc-editor.org/info/rfc2812>>.
- [RFC3233] Hoffman, P. and S. Bradner, "Defining the IETF", BCP 58, RFC 3233, DOI 10.17487/RFC3233, February 2002, <<https://www.rfc-editor.org/info/rfc3233>>.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <<https://www.rfc-editor.org/info/rfc4033>>.
- [RFC4084] Klensin, J., "Terminology for Describing Internet Connectivity", BCP 104, RFC 4084, DOI 10.17487/RFC4084, May 2005, <<https://www.rfc-editor.org/info/rfc4084>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC4880] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, DOI 10.17487/RFC4880, November 2007, <<https://www.rfc-editor.org/info/rfc4880>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.

- [RFC5694] Camarillo, G., Ed. and IAB, "Peer-to-Peer (P2P) Architecture: Definition, Taxonomies, Examples, and Applicability", RFC 5694, DOI 10.17487/RFC5694, November 2009, <<https://www.rfc-editor.org/info/rfc5694>>.
- [RFC5751] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, DOI 10.17487/RFC5751, January 2010, <<https://www.rfc-editor.org/info/rfc5751>>.
- [RFC6176] Turner, S. and T. Polk, "Prohibiting Secure Sockets Layer (SSL) Version 2.0", RFC 6176, DOI 10.17487/RFC6176, March 2011, <<https://www.rfc-editor.org/info/rfc6176>>.
- [RFC7118] Baz Castillo, I., Millan Villegas, J., and V. Pascual, "The WebSocket Protocol as a Transport for the Session Initiation Protocol (SIP)", RFC 7118, DOI 10.17487/RFC7118, January 2014, <<https://www.rfc-editor.org/info/rfc7118>>.
- [RFC7194] Hartmann, R., "Default Port for Internet Relay Chat (IRC) via TLS/SSL", RFC 7194, DOI 10.17487/RFC7194, August 2014, <<https://www.rfc-editor.org/info/rfc7194>>.
- [RFC7754] Barnes, R., Cooper, A., Kolkman, O., Thaler, D., and E. Nordmark, "Technical Considerations for Internet Service Blocking and Filtering", RFC 7754, DOI 10.17487/RFC7754, March 2016, <<https://www.rfc-editor.org/info/rfc7754>>.
- [RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <<https://www.rfc-editor.org/info/rfc7858>>.
- [RFC8280] ten Oever, N. and C. Cath, "Research into Human Rights Protocol Considerations", RFC 8280, DOI 10.17487/RFC8280, October 2017, <<https://www.rfc-editor.org/info/rfc8280>>.
- [RPZ] Vixie, P. and V. Schyver, "DNS Response Policy Zones (RPZ)", 2017, <<https://tools.ietf.org/html/draft-ietf-dnsop-dns-rpz-00>>.
- [SchengenRouting] Wikipedia, ., "Schengen Routing", 2018, <https://en.wikipedia.org/wiki/Schengen_Routing>.

- [Schleuder] Nadir, "Schleuder - A gpg-enabled mailinglist with remailing-capabilities.", 2017, <<https://schleuder.nadir.org/>>.
- [Sink] Sink, E., "Version Control by Example", 2011, <<http://ericsink.com/vcbe/>>.
- [Star] Star, S., "The Ethnography of Infrastructure", American Behavioral Scientist, Volume 43 (3), 377-391. , 1999, <<http://journals.sagepub.com/doi/abs/10.1177/00027649921955326>>.
- [StarRuhleder] Star, S. and K. Ruhleder, "Steps toward an ecology of infrastructure: Design and access for large information spaces", Information Systems Research 7 (1) (1996) 111-134. , 1996.
- [Swire] Peter Swire, ., "Social Networks, Privacy, and Freedom of Association: Data Empowerment vs. Data Protection", North Carolina Law Review (2012) 90 (1): 104. , 2012, <<https://ssrn.com/abstract=1989516> or <http://dx.doi.org/10.2139/ssrn.1989516>>.
- [Tocqueville] de Tocqueville, A., "Democracy in America", 1840, <http://classiques.uqac.ca/classiques/De_tocqueville_alexis/democracy_in_america_historical_critical_ed/democracy_in_america_vol_2.pdf p. 304>.
- [Troncosoetal] Troncoso, C., Isaakdis, M., Danezis, G., and H. Halpin, "Systematizing Decentralization and Privacy: Lessons from 15 Years of Research and Deployments", Proceedings on Privacy Enhancing Technologies ; 2017 (4):307-329 , 2017, <<https://www.petsymposium.org/2017/papers/issue4/paper87-2017-4-source.pdf>>.
- [Tufekci] Tufekci, Z., "Twitter and Tear Gas: The Power and Fragility of Networked Protest", 2017, <<https://www.twitterandteargas.org/>>.
- [UDHR] United Nations General Assembly, "The Universal Declaration of Human Rights", 1948, <<http://www.un.org/en/documents/udhr/>>.

- [UNGA] Hina Jilani, ., "Human rights defenders", A/59/401 , 2004, <http://www.un.org/en/ga/search/view_doc.asp?symbol=A/59/401 para. 46>.
- [UNHRC] Maina Kiai, ., "Report of the Special Rapporteur on the rights to freedom of peaceful assembly and of association", A/HRC/20/27 , 2012, <http://freeassembly.net/wp-content/uploads/2013/10/A-HRC-20-27_en-annual-report-May-2012.pdf>.
- [Vu] Vu, Quang Hieu, ., Lupu, Mihai, ., and . Ooi, Beng Chin, "Peer-to-Peer Computing: Principles and Applications", 2010, <<https://www.springer.com/cn/book/9783642035135>>.
- [Weiser] Weiser, L., "The Computer for the 21st Century", Scientific American Ubicomp Paper after Sci Am editing , 1991, <<https://web.archive.org/web/20141022035044/http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html>>.
- [Wugh] Nottingham, M., "Using Third Party Services for IETF Work", 2017, <<https://datatracker.ietf.org/doc/draft-nottingham-wugh-services/>>.

13.2. URIs

- [1] <mailto:hrpc@ietf.org>
- [2] <https://www.irtf.org/mailman/listinfo/hrpc>
- [3] <https://www.irtf.org/mail-archive/web/hrpc/current/index.html>

Authors' Addresses

Niels ten Oever
University of Amsterdam

EMail: mail@nielstenoever.net

Gisela Perez de Acha
Derechos Digitales

EMail: gisela@derechosdigitales.org

Human Rights Protocol Considerations Research Group
Internet-Draft
Intended status: Informational
Expires: December 20, 2018

N. ten Oever
University of Amsterdam
A. Andersdotter
ARTICLE 19
June 18, 2018

On the Politics of Standards
draft-tenoever-hrpc-political-05

Abstract

The IETF cannot ordain which standards or protocols are to be used on network, but the standards developing process in the IETF has a normative effect. Among other things the standardisation work at the IETF has implications on what is perceived as technologically possible and useful where networking technologies are being deployed, and its standards output reflect what is considered by the technical community as feasible and good practice. Because it mediates many aspects of modern life, and therefore contributes to the ordering of societies and communities, the consideration of the politics and (potential) impact of protocols should be part of the standardization and development process.

Status of This Memo

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

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

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

This Internet-Draft will expire on December 20, 2018.

Copyright Notice

Copyright (c) 2018 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

(<https://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. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Vocabulary Used	3
3. Research Question	4
4. Technology and Politics: a literature review	4
4.1. Technology is value neutral	4
4.2. Some protocols are political some times	5
4.3. All protocols are political sometimes	5
4.4. The network has its own logic and values	5
4.5. Protocols are inherently political	6
5. IETF: Protocols as Standards	7
5.1. Competition and collaboration	8
5.2. IETF standards setting externalities	9
5.2.1. Finance	9
5.2.2. Interoperability and backward compatability	9
5.2.3. Competition between layers	9
5.3. How voluntary are open standards?	10
6. The need for a positioning	10
7. Conclusion	11
8. The way forward	11
9. Security Considerations	12
10. IANA Considerations	12
11. Acknowledgements	12
12. Research Group Information	12
13. References	12
13.1. Informative References	12
13.2. URIs	17
Authors' Addresses	17

1. Introduction

"Science and technology lie at the heart of social asymmetry. Thus technology both creates systems which close off other options and generate novel, unpredictable and indeed previously unthinkable, option. The game of technology is never finished, and its ramifications are endless.

- Michel Callon

The design of the Internet through protocols and standards is a technical issue with great political and economic impacts [RFC0613]. The early Internet community already realized that it needed to make decisions on political issues such as Intellectual Property, Internationalization [BramanI], diversity, access [RFC0101] privacy and security [RFC0049], and the military [RFC0164] [RFC0316], governmental [RFC0144] [RFC0286] [RFC0313] [RFC0542] [RFC0549] and non-governmental [RFC0196] uses, which has been clearly pointed out by Braman [BramanII].

Recently there has been an increased discussion on the relation between Internet protocols and human rights [RFC8280] which spurred the discussion on the political nature of standards. The network infrastructure is on the one hand designed, described, developed, standardized and implemented by the Internet community, but the Internet community and Internet users are also shaped by the affordances of the technology. Companies, citizens, governments, standards developing bodies, public opinion and public interest groups all play a part in these discussions. In this document we aim to outline different views on the relation between standards and politics and seek to answer the question whether standards are political, and if so, how.

2. Vocabulary Used

Politics (from Greek: Politika: Politika, definition "affairs of the commons") is the process of making decisions applying to all members of a diverse group with conflicting interests. More narrowly, it refers to achieving and exercising positions of governance or organized control over a community. Furthermore, politics is the study or practice of the distribution of power and resources within a given community as well as the interrelationship(s) between communities. (adapted from [HagueHarrop])

Affordances The possibilities that are provided to an actors through the ordering of an environment by a technology.

Protocols 'Protocols are rules governing communication between devices or applications, and the creation or manipulation of any logical or communicative artifacts concomitant with such communication.' [Sisson]

Standards 'An Internet Standard is a specification that is stable and well-understood, is technically competent, has multiple, independent, and interoperable implementations with substantial operational experience, enjoys significant public support, and is

recognizably useful in some or all parts of the Internet.'
[RFC2026]

3. Research Question

Are protocols political? If so, should the politics of protocols need to be taken into account in their development process?

4. Technology and Politics: a literature review

In 1993 the Computer Professionals for Social Responsibility stated that 'the Internet should meet public interest objectives', similarly [RFC3935] states that 'The Internet isn't value-neutral, and neither is the IETF.'. Ethics and the Internet was already a topic of an RFC by the IAB in 1989 [RFC1097]. Nonetheless there has been a recent uptick in discussions around the impact of Internet protocols on human rights [RFC8280] in the IETF and more general about the impact of technology on society in the public debate.

This document aims to provide an overview of the spectrum of different positions that have been observed in the IETF and IRTF community, during participatory observation, through 39 interviews with members of the community, the Human Rights Protocol Considerations Research Group mailinglist and during and after the Technical Plenary on Protocols and Human Rights during IETF98. Without judging them on their internal or external consistency they are represented here, where possible we sought to engage with academic literature on this topic.

4.1. Technology is value neutral

This position starts from the premise that the technical and political are differentiated fields and that technology is 'value free'. This is also put more explicitly by Carey: "electronics is neither the arrival of apocalypse nor the dispensation of grace. Technology is technology; it is a means for communication and transportation over space, and nothing more." [Carey]. In this view protocols only become political when it is actually being used by humans. So the technology itself is not political, the use of the technology is. This view sees technology as instrument; "technologies are 'tools' standing ready to serve the purposes of their users. Technology is deemed 'neutral,' without evaluative content of its own.'" [Feenberg]. Feenberg continues: "technology is not inherently good or bad, and can be used to whatever political or social ends desired by the person or institution in control. Technology is a 'rational entity' and universally applicable. One may make exceptions on moral grounds, but one must also understand

that the "price for the achievement of environmental, ethical, or religious goals...is reduced efficiency." [Feenberg].

4.2. Some protocols are political some times

This stance is a pragmatic approach to the problem. It states that some protocols under certain conditions can themselves have a political dimension. This is different from the claim that a protocol might sometimes be used in a political way; that view is consistent with the idea of the technology being neutral (for the human action using the technology is where the politics lies). Instead, this position requires that each protocol and use be evaluated for its political dimension, in order to understand the extent to which it is political.

4.3. All protocols are political sometimes

While not an absolutist standpoint it recognizes that all design decisions are subject to the law of unintended consequences. The system consisting of the Internet and its users is vastly too complex to be predictable; it is chaotic in nature; its emergent properties cannot be predicted. This concept strongly hinges on the general purpose aspect of information technology and its malleability. Whereas not all (potential) behaviours, affordances and impacts of protocols can possible be predicted, one could at least consider the impact of proposed implementations.

4.4. The network has its own logic and values

While humans create technologies, this does not mean that they are forever under human control. A technology, once created, has its own logic that is independent of the human actors that either create or use the technology.

From this perspective, technologies can shape the world. As Martin Heidegger says, "The hydroelectric plant is not built into the Rhine River as was the old wooden bridge that joined bank with bank for hundreds of years. Rather the river is dammed up into the power plant. What the river is now, namely, a water power supplier, derives from out of the essence of the power station." [Heidegger] (p 16) The dam in the river changes the world in a way the bridge does not, because the dam alters the nature of the river.

In the same way -in another and more recent example- the very existence automobiles impose physical forms on the world different from those that come from the electric tram or the horse-cart. The logic of the automobile means speed and the rapid covering of distance, which encourages suburban development and a tendency toward

conurbation. But even if that did not happen, widespread automobile use requires paved roads, and parking lots and structures. These are pressures that come from the automotive technology itself, and would not arise without that technology.

In much same way, then, networking technology, such as protocols, creates its own demands. One of the most important conditions for protocol success is its incremental deployability [RFC5218]. This means that the network already contains constraints on what can be deployed into it. In this sense the network creates its own paths, but also has its own objective. According to this view the goal of the network is interconnection and connectivity; more connectivity is good for the network. Proponents of this positions also often describe the Internet as an organism with its own unique ecosystem.

In this position it is not necessarily clear where the 'social' ends and the 'technical' begins, and it could be argued that the distinction itself is a social construction [BijkerLaw] or that a real-life distinction between the two is hard to be made [Bloor].

4.5. Protocols are inherently political

This position argues the opposite of 'technological neutrality'. This position can be illustrated with Postman where he writes: 'the uses made of technology are largely determined by the structure of the technology itself' [Postman]. He states that the medium itself 'contains an ideological bias'. He continues to argue that technology is non-neutral:

(1) because of the symbolic forms in which information is encoded, different media have different intellectual and emotional biases; (2) because of the accessibility and speed of their information, different media have different political biases; (3) because of their physical form, different media have different sensory biases; (4) because of the conditions in which we attend to them, different media have different social biases; (5) because of their technical and economic structure, different media have different content biases.

Recent scholars of Internet infrastructure and governance have also pointed out that Internet processes and standards have become part and parcel of political processes and public policies. Several concrete examples are found within this approach, for instance, the IANA transition or global innovation policy [DeNardis]. The Raven process in which the IETF refused to standardize wiretapping -which resulted in [RFC2804]- was an instance where an international governance body took a position that was largely political, although driven by a technical argument. The process that led to [RFC6973] is similar: the Snowden disclosures which occurred in the political

space, engendered the IETF to act. This is summarized in [Abbate] who says: "protocols are politics by other means", emphasizing the interests that are at play in the process of designing standards.

This position further holds that protocols can never be understood without their contextual embeddedness: protocols do not exist solely by themselves but always are to be understood in a more complex context - the stack, hardware, or nation-state interests and their impact on civil rights. Finally, this view is that that protocols are political because they affect or sometimes effect the socio-technical ordering of reality. The latter observation leads Winner to conclude that the reality of technological progress has too often been a scenario where the innovation has dictated change for society. Those who had the power to introduce a new technology also had the power to create a consumer class to use the technology 'with new practices, relationships, and identities supplanting the old, --and those who had the wherewithal to implement new technologies often molded society to match the needs of emerging technologies and organizations.' [Winner].

5. IETF: Protocols as Standards

In the previous section we gave an overview of the different existing positions of the impact of Internet protocols in the Internet community. In the following section we will consider the standards setting process and its consequences for the politics of protocols.

Standards enabling interoperating networks, what we think of today as the Internet, were created as open, formal and voluntary standards. A platform for internet standardisation, the Internet Engineering Task Force (IETF), was created in 1992 to enable the continuation of such standardisation work. The IETF has sought to make the standards process transparent (by ensuring everyone can access standards, mailing-lists and meetings), predictable (by having clear procedures and reviews) and of high quality (by having draft documents reviewed by members from its own epistemic community). This is all aimed at increasing the accountability of the process and the quality of the standard.

The IETF implements what has been referred to as an "informal ex ante disclosure policy" for patents [Contreras], which includes the possibility for participants to disclose the existence of a patent relevant for the standard, royalty-terms which would apply to the implementors of that standard should it enter into effect, as well as other licensing terms that may be interesting for implementors to know. The community ethos in the IETF seems to lead to 100% royalty-free disclosures of prior patents which is a record number, even among other comparable standard organisations [Contreras].

5.1. Competition and collaboration

Standards exist for nearly everything: processes, technologies, safety, hiring, elections, and training. Standards provide blue-prints for how to accomplish a particular task in a similar way for others that are trying to accomplish the same thing, while reducing overhead and inefficiencies. Although there are different types and configurations of standards, they all enhance competition by allowing different entities to work from a commonly accepted baseline.

On the first types of standards than can be found are "informal" ones -agreed upon normal ways of interacting within a specific community. For example, the process through which greetings to a new acquaintance are expressed through a bow, a handshake or a kiss. On the other hand "formal" standards, are normally codified in writing. The next subsection will ---

Within economy studies, *_de facto_* standards arise in market situations where one entity is particularly dominant; downstream competitors are therefore tied to the dominant entity's technological solutions [Ahlborn]. Under EU anti-trust law, *de facto* standards have been found to restrict competition for downstream services in PC software products [CJEU2007], as well as downstream services dependent on health information [CJEU2004].

Even in international law, the World Trade Organisation (WTO) uses standards, although it recognises a difference between standards and technical regulations. The former are voluntary formal codes to which products or services may conform, while technical regulations are mandatory requirements to be fulfilled for a product to be accessible on one of the WTO country markets. These rules have implications for how nation states bounded by the WTO agreements can impose specific technical requirements on companies. Nonetheless, there are many standardisation groups that were originally launched by nation states or groups of nation states. ISO, BIS, CNIS, NIST, ABNT and ETSI are examples of institutions that are, wholly or partially, sponsored by public money in order to ensure smooth development of formal standards. Even if under WTO rules these organisations cannot create the equivalent of a technical regulation, they have important normative functions in their respective countries. No matter what form, all standards enhance competition and collaboration because they define a common approach to a problem. This potentially allows different instances to interoperate or be evaluated according to the same indicators.

The development of formal standards faces a number of economic and organisational challenges. Mainly, the cost and difficulty of organising many entities around a mutual goal, as well as the cost of

research and development leading up to a mutually beneficial technological platform. In addition, deciding what the mutual goal is can also be a problem. These challenges may be described as inter-organisational costs. Even after a goal is decided upon, coordination of multiple entities requires time and money. One needs communication platforms, processes and a commitment to mutual investment in a higher good. They are not simple tasks, and the more different communities are affected by a particular standardisation process, the more difficult the organisational challenges become.

5.2. IETF standards setting externalities

In spite of a strong community ethos and transparent procedures, the IETF is not immune to externalities.

5.2.1. Finance

Sponsorship to the IETF is varied, but is also of the nature that ongoing projects that are in the specific interest of one or some group of corporations may be given more funding than other projects (see [draft-finance-thoughts]). The IETF has faced three periods of decreased commitment from participants in funding its meetings in the past ten years, leading, naturally, to self-scrutiny, see for instance [IAOC69], [IAOC77], [IAOC99].

5.2.2. Interoperability and backward compatability

The need for interoperability, and backward compatability makes engineering work harder. And once a standard is designed, it does not automatically mean it will be broadly adopted at a fast pace. Examples of this are IPv6, DNSSEC, DKIM, etc. The need for interoperability means that a new protocol needs to take into account a much more diverse environment than early protocols, and also be amendable to different needs: protocols needs to relate and negotiate in a busy agora, as do the protocol developers. This means that some might get priority, whereas others get dropped.

5.2.3. Competition between layers

There is a competition between layers, and even contestation about what the borders of different layers are. This leads to competition between layers and different solutions for similar problems on different layers, which in its turn leads to further ossification, which leads to more contestation.

5.3. How voluntary are open standards?

Coordinating transnational stakeholders in a process of negotiation and agreement through the development of common rules is a form of global governance [Nadvi]. Standards are among the mechanisms by which this governance is achieved. Conformance to certain standards is often a basic condition of participation in international trade and communication, so there are strong economic and political incentives to conform, even in the absence of legal requirements [Russell]. [RogersEden] argue:

"As unequal participants compete to define standards, technological compromises emerge, which add complexity to standards. For instance, when working group participants propose competing solutions, it may be easier for them to agree on a standard that combines all the proposals rather than choosing any single proposal. This shifts the responsibility for selecting a solution onto those who implement the standard, which can lead to complex implementations that may not be interoperable. On its face this appears to be a failure of the standardization process, but this outcome may benefit certain participants-- for example, by allowing an implementer with large market share to establish a de facto standard within the scope of the documented standard."

6. The need for a positioning

It is indisputable that the Internet plays an increasingly important role in the lives of individuals. The community that produces standards for the Internet therefore also has an impact on society, which it itself has recognised in a number of previously adopted documents [RFC1958].

The IETF cannot ordain which standards are to be used on the networks, and it specifically does not determine the laws of regions or countries where networks are being used, but it does set open standards for interoperability on the Internet, and has done so since the inception of the Internet. Because a standard is the blue-print for how to accomplish a particular task in a similar way to others, the standards adopted have a normative effect. The standardisation work at the IETF will have implications on what is perceived as technologically possible and useful where networking technologies are being deployed, and its standards output reflect what is considered by the technical community as feasible and good practice.

This calls for providing a methodology in the IETF community to evaluate which routes forward should indeed be feasible, what constitutes the "good" in "good practice" and what trade-offs between different feasible features of technologies are useful and should

therefore be made possible. Such an analysis should take societal implication into account.

The risk of not doing this is threefold: (1) the IETF might make decisions which have a political impact that was not intended by the community, (2) other bodies or entities might make the decisions for the IETF because the IETF does not have an explicit stance, (3) other bodies that do take these issues into account might increase in importance to the detriment of the influence of the IETF.

This does not mean the IETF does not have a position on particular political issues. The policies for open and diverse participation [RFC7704], the anti-harassment policy [RFC7776], as well as the Guidelines for Privacy Considerations [RFC6973] are proof of this. Nonetheless, these are all examples of positions about the IETF's work processes or product. What is absent is a way for IETF participants to evaluate their role with respect to the wider implications of that IETF work.

7. Conclusion

Economics, competition, collaboration, openness, and political impact have been an inherent part of the work of the IETF since its early beginnings, by its nature as standards developing organization, through the contributions of the members of the Internet community, and because the ordering effect the Internet has on society. Whereas there might not be agreement in the Internet community on what the specific political nature is of technological development, it is undisputed that standards and protocols are both product of a political process, and they can also be used for political means. Whereas there is no need for a unified philosophy of Internet protocols, it is in the benefit of the IETF, the Internet and arguably society at large to take this into account in the standards development process.

8. The way forward

There are instruments that can help the IETF develop an approach to address the politics of standards. Part of this can be found in [RFC8280] as well as the United National Guiding Principles for Business and Human Rights [UNGP]. But there is not a one-size-fits-all solution. The IETF is a particular organization, with a particular mandate, and even if a policy is in place, its success depends on the implementation of the policy by the community.

Since 'de facto standardization is reliant on market forces' [Hanseth] we need to live with the fact standards bodies have a political nature [Webster]. This does not need to be problematic as

long as there are sufficient accountability and transparency mechanisms in place. The importance of these mechanisms increases with the importance of the standards and their implementations. The complexity of the work inscribes a requirement of competence in the work in the IETF, which forms an inherent barrier for end-user involvement. Even though this might not be intentional, it is a result of the interplay between the characteristics of the epistemic community in the IETF and the nature of the standard setting process.

Instead of splitting hairs about whether 'standards are political' [Winner] [Woolgar] we argue that we need to look at the politics of individual standards and invite document authors and reviewers to take these dynamics into account.

9. Security Considerations

As this draft concerns a research document, there are no security considerations as described in [RFC3552], which does not mean that not addressing the issues brought up in this draft will not impact the security of end-users or operators.

10. IANA Considerations

This document has no actions for IANA.

11. Acknowledgements

Thanks to Andrew Sullivan, Brian Carpenter, Mark Perkins and all contributors and reviewers on the hrpc mailinglist. Special thanks to Gisela Perez de Acha for some thorough editing rounds.

12. Research Group Information

The discussion list for the IRTF Human Rights Protocol Considerations working group is located at the e-mail address hrpc@ietf.org [1]. Information on the group and information on how to subscribe to the list is at: <https://www.irtf.org/mailman/listinfo/hrpc> [2]

Archives of the list can be found at: <https://www.irtf.org/mail-archive/web/hrpc/current/index.html> [3]

13. References

13.1. Informative References

[Abbate] Abbate, J., "Inventing the Internet", MIT Press , 2000, <<https://mitpress.mit.edu/books/inventing-internet>>.

- [Ahlborn] Ahlborn, C., Denicolo, V., Geradin, D., and A. Padilla, "Implications of the Proposed Framework and Antitrust Rules for Dynamically Competitive Industries", DG Comp's Discussion Paper on Article 82, DG COMP, European Commission , 2006, <<http://curia.europa.eu/juris/liste.jsf?num=T-201/04>>.
- [BijkerLaw] Bijker, W. and J. Law, "Shaping Technology/ Building Society: Studies in Sociotechnical Change", Cambridge, MA: MIT Press , 1992.
- [Bloor] Bloor, D., "Knowledge and Social Imagery", London: Routeledge & Kegan Paul , 1976.
- [BramanI] Braman, S., "Internationalization of the Internet by design: The first decade", Global Media and Communication, Vol 8, Issue 1, pp. 27 - 45 , 2012, <<http://dx.doi.org.proxy.uba.uva.nl:2048/10.1177%2F1742766511434731>>.
- [BramanII] Braman, S., "The Framing Years: Policy Fundamentals in the Internet Design Process, 1969-1979", The Information Society Vol. 27, Issue 5, 2011 , 2010, <<http://dx.doi.org.proxy.uba.uva.nl:2048/10.1080/01972243.2011.607027>>.
- [Carey] Carey, J., "Communication As Culture", p. 139 , 1992.
- [CJEU2004] Court of Justice of the European Union, ., "ECLI:EU:C:2004:257, C-418/01 IMS Health", Cambridge, UK: Cambridge University Press , 2004, <<http://curia.europa.eu/juris/liste.jsf?num=C-418/01>>.
- [CJEU2007] Court of Justice of the European Union, ., "ECLI:EU:T:2007:289, T-201/04 Microsoft Corp.", Cambridge, UK: Cambridge University Press , 2007, <<http://curia.europa.eu/juris/liste.jsf?num=T-201/04>>.
- [Contreras] Contreras, J., "Technical Standards and Ex Ante Disclosure: Results and Analysis of an Empirical Study", Jurimetrics: The Journal of Law, Science & Technology, vol. 53, p. 163-211 , 2013.

- [DeNardis] Denardis, L., "The Internet Design Tension between Surveillance and Security", IEEE Annals of the History of Computing (volume 37-2) , 2015, <<http://is.gd/7GANFy>>.
- [draft-finance-thoughts] Arkko, J., "Thoughts on IETF Finance Arrangements", 2017, <<https://datatracker.ietf.org/doc/html/draft-arkko-ietf-finance-thoughts>>.
- [Feenberg] Feenberg, A., "Critical Theory of Technology", p.5-6 , 1991.
- [HagueHarrop] Hague, R. and M. Harrop, "Comparative Government and Politics: An Introduction", Macmillan International Higher Education. pp. 1-. ISBN 978-1-137-31786-5. , 2013.
- [Hanseth] Hanseth, O. and E. Monteiro, "Inscribing Behaviour in Information Infrastructure Standards", Accounting, Management and Information Technology 7 (14) p.183-211 , 1997.
- [Heidegger] Heidegger, M., "The Question Concerning Technology and Other Essays", Garland: New York, 1977 , 1977, <http://ssbothwell.com/documents/ebooksclub.org__The_Question_Concerning_Technology_and_Other_Essays.pdf>.
- [IAOC69] IAOC, ., "IAOC Report Chicago", 2007, <<https://iaoc.ietf.org/documents/IAOC-Report-Chicago-69.pdf>>.
- [IAOC77] IAOC, ., "IAOC Report Anaheim", 2010, <<https://iaoc.ietf.org/documents/IAOC-Report-Anaheim-77.pdf>>.
- [IAOC99] IAOC, ., "IAOC Report Prague", 2017, <<https://iaoc.ietf.org/documents/IAOCReportinAdvanceofIETF99.pdf>>.
- [Nadvi] Nadvi, K. and F. Waeltring, "Making sense of global standards", In: H. Schmitz (Ed.), Local enterprises in the global economy (pp. 53-94). Cheltenham, UK: Edward Elgar. , 2004.

- [Postman] Postman, N., "Technopoly: the Surrender of Culture to Technology", Vintage: New York. pp. 3-20. , 1992.
- [RFC0049] Meyer, E., "Conversations with S. Crocker (UCLA)", RFC 49, DOI 10.17487/RFC0049, April 1970, <<https://www.rfc-editor.org/info/rfc49>>.
- [RFC0101] Watson, R., "Notes on the Network Working Group meeting, Urbana, Illinois, February 17, 1971", RFC 101, DOI 10.17487/RFC0101, February 1971, <<https://www.rfc-editor.org/info/rfc101>>.
- [RFC0144] Shoshani, A., "Data sharing on computer networks", RFC 144, DOI 10.17487/RFC0144, April 1971, <<https://www.rfc-editor.org/info/rfc144>>.
- [RFC0164] Heafner, J., "Minutes of Network Working Group meeting, 5/16 through 5/19/71", RFC 164, DOI 10.17487/RFC0164, May 1971, <<https://www.rfc-editor.org/info/rfc164>>.
- [RFC0196] Watson, R., "Mail Box Protocol", RFC 196, DOI 10.17487/RFC0196, July 1971, <<https://www.rfc-editor.org/info/rfc196>>.
- [RFC0286] Forman, E., "Network Library Information System", RFC 286, DOI 10.17487/RFC0286, December 1971, <<https://www.rfc-editor.org/info/rfc286>>.
- [RFC0313] O'Sullivan, T., "Computer based instruction", RFC 313, DOI 10.17487/RFC0313, March 1972, <<https://www.rfc-editor.org/info/rfc313>>.
- [RFC0316] McKay, D. and A. Mullery, "ARPA Network Data Management Working Group", RFC 316, DOI 10.17487/RFC0316, February 1972, <<https://www.rfc-editor.org/info/rfc316>>.
- [RFC0542] Neigus, N., "File Transfer Protocol", RFC 542, DOI 10.17487/RFC0542, August 1973, <<https://www.rfc-editor.org/info/rfc542>>.
- [RFC0549] Michener, J., "Minutes of Network Graphics Group meeting, 15-17 July 1973", RFC 549, DOI 10.17487/RFC0549, July 1973, <<https://www.rfc-editor.org/info/rfc549>>.
- [RFC0613] McKenzie, A., "Network connectivity: A response to RFC 603", RFC 613, DOI 10.17487/RFC0613, January 1974, <<https://www.rfc-editor.org/info/rfc613>>.

- [RFC1097] Miller, B., "Telnet subliminal-message option", RFC 1097, DOI 10.17487/RFC1097, April 1989, <<https://www.rfc-editor.org/info/rfc1097>>.
- [RFC1958] Carpenter, B., Ed., "Architectural Principles of the Internet", RFC 1958, DOI 10.17487/RFC1958, June 1996, <<https://www.rfc-editor.org/info/rfc1958>>.
- [RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, DOI 10.17487/RFC2026, October 1996, <<https://www.rfc-editor.org/info/rfc2026>>.
- [RFC2804] IAB and IESG, "IETF Policy on Wiretapping", RFC 2804, DOI 10.17487/RFC2804, May 2000, <<https://www.rfc-editor.org/info/rfc2804>>.
- [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <<https://www.rfc-editor.org/info/rfc3552>>.
- [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>>.
- [RFC5218] Thaler, D. and B. Aboba, "What Makes for a Successful Protocol?", RFC 5218, DOI 10.17487/RFC5218, July 2008, <<https://www.rfc-editor.org/info/rfc5218>>.
- [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>>.
- [RFC7704] Crocker, D. and N. Clark, "An IETF with Much Diversity and Professional Conduct", RFC 7704, DOI 10.17487/RFC7704, November 2015, <<https://www.rfc-editor.org/info/rfc7704>>.
- [RFC7776] Resnick, P. and A. Farrel, "IETF Anti-Harassment Procedures", BCP 25, RFC 7776, DOI 10.17487/RFC7776, March 2016, <<https://www.rfc-editor.org/info/rfc7776>>.
- [RFC8280] ten Oever, N. and C. Cath, "Research into Human Rights Protocol Considerations", RFC 8280, DOI 10.17487/RFC8280, October 2017, <<https://www.rfc-editor.org/info/rfc8280>>.

[RogersEden]

Rogers, M. and G. Eden, "The Snowden Disclosures, Technical Standards, and the Making of Surveillance Infrastructures", *International Journal of Communication* 11(2017), 802-823 , 2017, <<http://ijoc.org/index.php/ijoc/article/view/5525/1941>>.

[Russell] Russell, A., "Open standards and the digital age: History, ideology, and networks", Cambridge, UK: Cambridge University Press , 2014.

[Sisson] Sisson, D., "Standards and Protocols", 2000, <<https://philosophe.com/design/standards/>>.

[UNGP] Ruggie, J. and United Nations, "United Nations Guiding Principles for Business and Human Rights", 2011, <http://www.ohchr.org/Documents/Publications/GuidingPrinciplesBusinessHR_EN.pdf>.

[Webster] Webster, J., "Networks of Collaboration or Conflict? The Development of EDI", *The social shaping of inter-organizational IT systems and data interchange*, eds: I. McLoughling & D. Mason, European Commission PICT/COST A4 , 1995.

[Winner] Winner, L., "Upon openig the black box and finding it empty: Social constructivism and the philosophy of technology", *Science, Technology, and Human Values* 18 (3) p. 362-378 , 1993.

[Woolgar] Woolgar, S., "Configuring the user: the case of usability trials", *A sociology of monsters. Essays on power, technology and dominatior*, ed: J. Law, Routeledge p. 57-102. , 1991.

13.2. URIs

[1] <mailto:hrpc@ietf.org>

[2] <https://www.irtf.org/mailman/listinfo/hrpc>

[3] <https://www.irtf.org/mail-archive/web/hrpc/current/index.html>

Authors' Addresses

Niels ten Oever
University of Amsterdam

EMail: mail@nielstenoever.net

Amelia Andersdotter
ARTICLE 19

EMail: amelia@article19.org