                  The Wire Image of a Network Protocol
                       draft-iab-wire-image-01

Abstract

   This document defines the wire image, an abstraction of the
   information available to an on-path non-participant in a networking
   protocol. This abstraction is intended to shed light on the
   implications on increased encryption has for network functions that
   use the wire image.

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 May 9, 2019.

1. Introduction

A protocol specification defines a set of behaviors for each
participant in the protocol: which lower-layer protocols are used for
which services, how messages are formatted and protected, which
participant sends which message when, how each participant should
respond to each message, and so on.

Implicit in a protocol specification is the information the protocol
radiates toward nonparticipant observers of the messages sent among
participants, often including participants in lower layer protocols.
Any information that has a clear definition in the protocol's message
format(s), or is implied by that definition, and is not
cryptographically confidentiality-protected can be unambiguously
interpreted by those observers. This information comprises the
protocol's wire image, which we define and discuss in this document.

It is the wire image, not the protocol's specification, that
determines how third parties on the network paths among protocol
participants will interact with that protocol.

The increasing deployment of transport-layer security [RFC8226] to
protect application-layer headers and payload, as well as the
definition and deployment of QUIC [QUIC], a transport protocol which
encrypts most of its own control information, bring new relevance to
this question. QUIC is, in effect, the first IETF-defined transport
protocol to take care of the minimization of its own wire image, to
prevent ossification and improve end-to-end privacy by reducing
information radiation.

The flipside of this trend is the impact of a less visible wire image
on various functions driven by third-party observation of the wire
image. In contrast to ongoing discussions about this tussle, this
draft treats the wire image as a pure abstraction, with the hope that
it can shed some light on these discussions.

2. Definition

The wire image of the set of protocols in use for a given
communication is the view of that set of protocols as observed by an
entity not participating in the communication. It is the sequence of
packets sent by each participant in the communication, including the
content of those packets and metadata about the observation itself:
the time at which each packet is observed, and the vantage point of
the observer.

3. Discussion

   This definition illustrates some important properties of the wire
   image.

   Key is that the wire image is not limited to merely "the unencrypted
   bits in the header". In particular, the metadata, such as sequences
   of interpacket timing and packet sizes, can also be used to infer
   other parameters of the behavior of the protocols in use, or to
   fingerprint protocols and/or specific implementations of those
   protocols; see Section 3.2.

   An important implication of this property is that a protocol which
   uses confidentiality protection for the headers it needs to operate
   can be deliberately designed to have a specified wire image that is
   separate from that machinery; see Section 4. Note that this is a
   capability unique to encrypted protocols. Parts of a wire image may
   also be made visible to devices on path, but immutable through end-
   to-end integrity protection; see Section 3.3.

   Portions of the wire image of a protocol stack that are neither
   confidentiality-protected nor integrity-protected are writable by
   devices on the path(s) between the endpoints using the protocols. A
   protocol with a wire image that is largely writable operating over a
   path with devices that understand the semantics of the protocol's
   wire image can modify it, in order to induce behaviors at the
   protocol's participants. TCP is one such protocol in the current
   Internet.

   The term "wire image" can be applied in different scopes: the wire
   image of a single packet refers to the information derivable from
   observing that one packet in isolation; the wire image of a single
   protocol refers to the information derivable from observing only the
   headers belonging to that protocol on a sequence of packets, in
   isolation from other protocols in use for a communication. See
   Section 3.1 for more.

   For a given packet observed at a given point in the network, the wire
   image contains information from the entire stack of protocols in use
   at that observation point. This implies that the wire image depends
   on the observer as well: each observer may see a slightly different
   image of the same communication.

   In this document, we assume that only information at the transport
   layer and above is delivered end-to-end, and focus on the "Internet"
   wire image: that portion of the wire image at the network layer and
   above. While confidentiality and integrity protection may be added
   at multiple layers in the stack, MAC-layer integrity and

confidentiality protection do not prevent modification by the devices
terminating those security associations, or by devices on different
segments of the path.

3.1. The Extent of the Wire Image

While we begin this definition as the properties of a sequence of
packets in isolation, this is not how wire images are typically used
by passive observers. A passive observer will generally consider the
union of all the information in the wire image in all the packets
generated by a given conversation.

Similarly, the wire image of a single protocol is rarely seen in
isolation. The dynamics of the application and network stacks on
each endpoint use multiple protocols for any higher level task. Most
protocols involving user content, for example, are often seen on the
wire together with DNS traffic; the information from the wire image
from each protocol in use for a given communication can be correlated
to infer information about the dynamics of the overlying application.

Information from protocol wire images is also not generally used on
its own, but is rather additionally correlated with other context
information available to the observer: e.g. information about other
communications engaged in by each endpoint, information about the
implementations of the protocols at each endpoint, information about
the network and internetwork topology near those endpoints, and so
on. This context can be used together with information from the wire
image to reach more detailed inferences about endpoint and end-user
behavior.

Note also that the wire image is multidimensional. This implies that
the name "image" is not merely metaphorical, and that general image
recognition techniques may be applicable to extracting patterns and
information from it.

3.2. Obscuring timing and sizing information

Cryptography can protect the confidentiality of a protocol's headers,
to the extent that forwarding devices do not need the
confidentiality-protected information for basic forwarding
operations. Ciphersuites and other transmission techniques designed
to prevent timing analysis can also be applied at the sender to
reduce the information content of the metadata portion of the wire
image. However, there are limits to these techniques. Packets
cannot be made smaller than their information content, sent faster
than processing time requirements at the sender allow, or transmitted
through the network faster than a factor less than one of the speed
of light. Since these techniques operate at the expense of bandwidth

efficiency and latency, they are also limited to the application's
tolerance for latency and bandwidth inefficiency.

3.3. Integrity Protection of the Wire Image

Adding end-to-end integrity protection to portions of the wire image
makes it impossible for on-path devices to modify them without
detection by the endpoints, which can then take action in response to
those modifications, making these portions of the wire image
effectively immutable. However, they can still be observed by
devices on path. This allows the creation of signals intended by the
endpoints solely for the consumption of these on-path devices.

Integrity protection can only practically be applied to the sequence
of bits in each packet, which implies that a protocol's visible wire
image cannot be made completely immutable in a packet-switched
network. Interarrival timings, for instance, cannot be easily
protected, as the observable delay sequence is modified as packets
move through the network and experience different delays on different
links. Message sequences are also not practically protectable, as
packets may be dropped or reordered at any point in the network, as a
consequence of the network's operation. Intermediate systems with
knowledge of the protocol semantics in the readable portion of the
wire image can also purposely delay or drop packets in order to
affect the protocol's operation.

4. Engineering the Wire Image

Understanding the nature of a protocol's wire image allows it to be
engineered. The general principle at work here, observed through
experience with deployability and non-deployability of protocols at
the network and transport layers in the Internet, is that all
observable parts of a protocol's wire image will eventually be used
by devices on path; consequently, changes or future extensions that
affect the observable part of the wire image become difficult or
impossible to deploy.

A network function which serves a purpose useful to its deployer will
use the information it needs from the wire image, and will tend to
get that information from the wire image in the simplest way
possible.

For example, consider the case of the ubiquitous TCP [RFC0793]
transport protocol. As described in [PATH-SIGNALS], several key in-
network functions have evolved to take advantage of implicit signals
in TCP's wire image, which, as TCP provides neither integrity or
confidentiality protection for its headers, is inseparable from its
internal operation. Some of these include:

o Determining return routability and consent: For example, TCP's
   wire image contains both an implicit indication that the sender of
   a packet is at least on the path toward its source address (in the
   acknowledgement number during the handshake), as well as an
   implicit indication that a receiving device consents to continue
   communication. These are used by stateful network firewalls.

o Measuring loss and latency: For example, examining the sequence of
   TCP's sequence and acknowledgement numbers, as well as the ECN
   [RFC3168] control bits allows the inference of congestion, loss
   and retransmission along the path. The sequence and
   acknowledgement numbers together with the timestamp option
   [RFC7323] allow the measurement of application-experienced
   latency.

During the design of a protocol, the utility of features such as
these should be considered. The protocol's wire image can be
designed to explicitly expose information to those network functions
deemed important by the designers. The wire image should expose as
little other information as possible.

However, even when information is explicitly provided to the network,
any information that is exposed by the wire image, even that
information not intended to be consumed by an observer, must be
designed carefully, as deployed network functions using that
information may render it immutable for future versions of the
protocol. For example, information needed to support decryption by
the receiving endpoint (cryptographic handshakes, sequence numbers,
and so on) may be used by devices along the path for their own
purposes.

4.1. Declaring Protocol Invariants

One potential approach to reduce the extent of the wire image that
will be used by devices on the path is to define a set of invariants
for a protocol during its development. Declaring a protocol's
invariants represents a promise made by the protocol's developers
that certain bits in the wire image, and behaviors observable in the
wire image, will be preserved through the specification of all future
versions of the protocol. QUIC's invariants [QUIC-INVARIANTS] are an
initial attempt to apply this approach to QUIC.

While static aspects of the wire image - bits with simple semantics
at fixed positions in protocol headers - can easily be made
invariant, different aspects of the wire image may be more or less
appropriate to define as invariants. For a protocol with a version
and/or extension negotiation mechanism, the bits in the header and
behaviors tied to those bits which implement version negotiation

should be made invariant. More fluid aspects of the wire image and
behaviors which are not necessary for interoperability are not
appropriate as invariants.

Parts of a protocol's wire image not declared invariant but intended
to be visible to devices on path should be protected against
"accidental invariance": the deployment of on-path devices over time
that make simplifying assumptions about the behavior of those parts
of the wire image, making new behaviors not meeting those assumptions
difficult to deploy. Integrity protection of the wire image may
itself help protect against accidental invariance, because read-only
wire images invite less meddling than path-writable wire images. The
techniques discussed in [USE-IT] may also be useful in further
preventing accidental invariance and ossification.

Likewise, parts of a protocol's wire image not declared invariant and
not intended to be visible to the path should be encrypted to protect
their confidentiality. When confidentiality protection is either not
possible or not practical, then, as above, the approaches discussed
in [USE-IT] may be useful in ossification prevention.

4.2. Trustworthiness of Engineered Signals

Since they are separate from the signals that drive an encrypted
protocol's mechanisms, the accuracy of integrity-protected signals in
an engineered wire image intended for consumption by the path may not
be verifiable by on-path devices; see [PATH-SIGNALS]. Indeed, any
two endpoints with a secret channel between them (in this case, the
encrypted protocol itself) may collude to change the semantics and
information content of these signals. This is an unavoidable
consequence of the separation of the wire image from the protocol's
operation afforded by confidentiality protection of the protocol's
headers.

5. Acknowledgments

Thanks to Martin Thomson, Stephen Farrell, Thomas Fossati, Ted
Hardie, Mark Nottingham, Tommy Pauly, and the membership of the IAB
Stack Evolution Program, for text, feedback, and discussions that
have improved this document.

This work is partially supported by the European Commission under
Horizon 2020 grant agreement no. 688421 Measurement and Architecture
for a Middleboxed Internet (MAMI), and by the Swiss State Secretariat
for Education, Research, and Innovation under contract no. 15.0268.
This support does not imply endorsement.

6. Informative References

[PATH-SIGNALS]
          Hardie, T., "Path Signals", draft-hardie-path-signals-03
          (work in progress), April 2018.

[QUIC]   Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed
          and Secure Transport", draft-ietf-quic-transport-16 (work
          in progress), October 2018.

[QUIC-INVARIANTS]
          Thomson, M., "Version-Independent Properties of QUIC",
          draft-ietf-quic-invariants-03 (work in progress), October
          2018.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7,
          RFC 793, DOI 10.17487/RFC0793, September 1981,
          <https://www.rfc-editor.org/info/rfc793>.

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition
          of Explicit Congestion Notification (ECN) to IP",
          RFC 3168, DOI 10.17487/RFC3168, September 2001,
          <https://www.rfc-editor.org/info/rfc3168>.

[RFC7323] Borman, D., Braden, B., Jacobson, V., and R.
          Scheffenegger, Ed., "TCP Extensions for High Performance",
          RFC 7323, DOI 10.17487/RFC7323, September 2014,
          <https://www.rfc-editor.org/info/rfc7323>.

[RFC8226] Peterson, J. and S. Turner, "Secure Telephone Identity
          Credentials: Certificates", RFC 8226,
          DOI 10.17487/RFC8226, February 2018,
          <https://www.rfc-editor.org/info/rfc8226>.

[USE-IT]   Thomson, M., "Long-term Viability of Protocol Extension
          Mechanisms", draft-thomson-use-it-or-lose-it-02 (work in
          progress), June 2018.

Authors' Addresses

   Brian Trammell
   ETH Zurich
   Gloriastrasse 35
   8092 Zurich
   Switzerland

   Email: ietf@trammell.ch

      Mirja Kuehlewind
      ETH Zurich
      Gloriastrasse 35
      8092 Zurich
      Switzerland

      Email: mirja.kuehlewind@tik.ee.ethz.ch