       Further Key Words for Use in RFCs to Indicate Requirement Levels

Abstract

   RFC 2119 defines a standard set of key words for describing
   requirements of a specification. Many IETF documents have found that
   these words cannot accurately capture the nuanced requirements of
   their specification. This document defines additional key words that
   can be used to address alternative requirements scenarios. Authors
   who follow these guidelines should incorporate this phrase near the
   beginning of their document:

   The key words "MUST (BUT WE KNOW YOU WON'T)", "SHOULD CONSIDER",
   "REALLY SHOULD NOT", "OUGHT TO", "WOULD PROBABLY", "MAY WISH TO",
   "COULD", "POSSIBLE", and "MIGHT" in this document are to be
   interpreted as described in RFC 6919.

Status of This Memo

Copyright Notice

Table of Contents

1. MUST (BUT WE KNOW YOU WON'T)

    The phrase "MUST (BUT WE KNOW YOU WON'T)" is used to indicate
    requirements that are needed to meet formal review criteria (e.g.,
    mandatory-to-implement security mechanisms), when these mechanisms
    are too inconvenient for implementers to actually implement.

    This phrase is frequently used in a contracted form in which the
    parenthetical is omitted. The parenthetical may also be moved later
    in the sentence for stylistic reasons. If the parenthetical is
    present, authors MUST provide a reason why they know implementors
    will not heed this instruction in the parenthetical, as in the
    example (BUT WE KNOW YOU WON'T). In the below example, we show a
    case from RFC 6120 where the original text omitted the parenthetical,
    and we have indicated an appropriate parenthetical.

    For example: "For authentication only, servers and clients MUST
    support the SASL Salted Challenge Response Authentication Mechanism
    [SCRAM] -- in particular, the SCRAM-SHA-1 and SCRAM-SHA-1-PLUS
    variants [(BUT WE KNOW YOU WON'T, because your TLS library doesn't
    support extracting channel binding information)]." [RFC6120]

2. SHOULD CONSIDER

    The phrase "SHOULD CONSIDER" indicates that the authors of the
    specification think that implementations should do something, but
    they're not sure quite what.

    For example: "Applications that take advantage of typed links should
    consider the attack vectors opened by automatically following,
    trusting, or otherwise using links gathered from HTTP headers."
    [RFC5988]

3. REALLY SHOULD NOT

    The phrase "REALLY SHOULD NOT" is used to indicate dangerous
    behaviors that some important vendor still does and therefore we were
    unable to make MUST NOT.

    For example: "This command really should not be used" [RFC0493]

4. OUGHT TO

   The phrase "OUGHT TO" conveys an optimistic assertion of an
   implementation behavior that is clearly morally right, and thus does
   not require substantiation.

   For example: "If a decision might affect semantic transparency, the
   implementor ought to err on the side of maintaining transparency
   unless a careful and complete analysis shows significant benefits in
   breaking transparency." [RFC2616]

5. WOULD PROBABLY

   The phrase "WOULD PROBABLY" indicates the authors expectation about
   what a reasonable implementation is likely to do in a given case.
   There is no requirement for implementations to be reasonable.

   This phrase is also a good example of an aspect of English grammar
   that is often useful in specification writing, namely the passive-
   aggressive voice, which provides a meaning in between the active and
   the passive voice.

   For example: "A SMTP client would probably only want to authenticate
   an SMTP server whose server certificate has a domain name that is the
   domain name that the client thought it was connecting to." [RFC3207]

6. MAY WISH TO

   The phrase "MAY WISH TO" indicates a behavior that might seem
   appealing to some people, but which is regarded as ridiculous or
   unnecessary by others. This phrase is frequently used to avoid
   further delay in approval of a document.

   For example: "Verifiers MAY wish to track testing mode results to
   assist the Signer." [RFC6376]

7. COULD

   The phrase "COULD" provides a way for specification authors to
   articulate existential possibilities, in order to provide a hint that
   might be critical to reliable or secure operation, but without a hard
   requirement. The lack of a requirement allows for vendor product
   differentiation.

   For example: "An implementation could mitigate this race condition,
   for example, using timers." [RFC6733]

8. POSSIBLE

   The phrase "POSSIBLE" describes what some of the working group
   members thought of as an edge case that will never happen, but in
   practice allows the protocol to work at the most fundamental level.

   For example: "It is also possible for the server to send a completion
   response for some other command (if multiple commands are in
   progress), or untagged data." [RFC3501]

9. MIGHT

   The phrase "MIGHT" conveys a requirement in an intentionally stealthy
   fashion, to facilitate product differentiation (cf. "COULD" above).

   For example: "In the case of audio and different "m" lines for
   different codecs, an implementation might decide to act as a mixer
   with the different incoming RTP sessions, which is the correct
   behavior." [RFC5888]

10. Security Considerations

   Traditionally, security requirements in IETF documents have been
   expressed with a mixture of requirements words from RFC 2119
   [RFC2119] and the phrases used above. The key words in RFC 2119 are
   principally useful when threats and mitigations are clear and well
   defined. The key words in this document can be applied when the
   threat model is ambiguous, and mitigations are unclear or
   inconvenient.

11. References

11.1. Normative References

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119, March 1997.

11.2. Informative References

   [RFC0493] Michener, J., Cotton, I., Kelley, K., Liddle, D., and E.
              Meyer, "GRAPHICS PROTOCOL", RFC 493, April 1973.

   [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H.,
              Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext
              Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

   [RFC3207] Hoffman, P., "SMTP Service Extension for Secure SMTP over
              Transport Layer Security", RFC 3207, February 2002.

   [RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION
              4rev1", RFC 3501, March 2003.

   [RFC5888] Camarillo, G. and H. Schulzrinne, "The Session Description
              Protocol (SDP) Grouping Framework", RFC 5888, June 2010.

   [RFC5988] Nottingham, M., "Web Linking", RFC 5988, October 2010.

   [RFC6120] Saint-Andre, P., "Extensible Messaging and Presence
              Protocol (XMPP): Core", RFC 6120, March 2011.

   [RFC6376] Crocker, D., Hansen, T., and M. Kucherawy, "DomainKeys
              Identified Mail (DKIM) Signatures", RFC 6376,
              September 2011.

   [RFC6733] Fajardo, V., Arkko, J., Loughney, J., and G. Zorn,
              "Diameter Base Protocol", RFC 6733, October 2012.

Authors' Addresses

   Richard Barnes
   BBN
   1300 N 17th St
   Arlington, VA 22209
   US

   EMail: rlb@ipv.sx

   Stephen Kent
   BBN
   10 Moulton St
   Cambridge, MA 02138
   US

   EMail: kent@bbn.com

   Eric Rescorla
   RTFM, Inc.
   2064 Edgewood Drive
   Palo Alto, CA 94303
   US

   EMail: ekr@rtfm.com