RFC3261 Interop Statement
draft-sparks-sip-3261-interop-statement-00

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on September 1, 2009.

Copyright Notice

Copyright (c) 2009 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 in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Abstract

This document captures an outline of the interoperability statements that will be collected to construct an interoperability report for RFC 3261. The outline is still under review and should not be treated
as complete, but will drive data collection at upcoming interoperability events.

Table of Contents

1. Documenting Interoperability to Inform Maintenance of SIP . . . 3
2. A proposed implementation report outline for RFC3261 . . . . 4
3. Open questions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
4. Security Considerations . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
5. Informative References . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
Author's Address . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
1. Documenting Interoperability to Inform Maintenance of SIP

To effectively maintain SIP [RFC3261], either through continued revision at Proposed Standard or advancement to Draft, we will have to be very careful to concentrate on identifying and documenting what is, and isn't, currently implemented and interoperable. To advance to Draft, [RFC2026] requires a report on the implementation and interoperability of the features in a standard. [I-D.dusseault-impl-reports] clarifies and refines that requirement. SIP has a very large number of normative statements in its specification (RFC 3261 alone has over one thousand [RFC2119] keyword-based requirements). Most of those statements interact. It is simply infeasible to produce a detailed checklist of every combination of every requirement along with a pair of implementations that have proven that configuration works. Rather, following the guidelines in [I-D.dusseault-impl-reports], what we can do is capture a set of higher level feature descriptions and agree that testing to those features will cover the full set of combinations of requirements. The interoperability report would capture pairs of implementations where that feature worked and call out any exceptions (requirements that were not met). An initial draft of what such a set of features for RFC3261 would look like appears in Section 2. A similar approach was taken for the interoperability report for RTP/RTCP (see http://www.ietf.org/IESG/Implementations/RTP-RTCP-Implementation.txt). While not required by process until advancing to Draft, such a report would provide useful information to guide where we focus our energy maintaining the standard at Proposed.

Additional reports may be required for some of the normatively referenced documents. It might be possible to cover some of those (particularly [RFC3263]) simply by including them in this report.
2. A proposed implementation report outline for RFC3261

- Basic mechanics
  - Interoperable completion of a non-INVITE transaction over an unreliable transport
  - Interoperable completion of an INVITE transaction over a reliable transport
  - Interoperable completion of an INVITE/200 through proxies with a mix of reliable and unreliable transport hops
  - Interoperable completion of a SIP transaction over
    - UDP
    - TCP
    - TLS
    - SCTP
  - Interoperable completion of SIP transactions over IPv6
  - Interoperable completion of a SIP transaction using non-default transaction state machine timers
  - Demonstrate correct implementation of handling "stray" responses
  - Demonstrate correct implementation of CANCEL at an endpoint (UAS and UAC)
  - Interoperable completion of a SIP transaction with messages using non-ascii UTF8
  - Demonstrate correct implementation of detection and recovery from INVITE glare
  - Demonstrate completion of a SIP transaction with messages using a mix of multiple comma-separated values after a header
name vs multiple instances of the header name
- Demonstrate successful completion of a SIP transaction using the "received" and "rport" mechanisms
- Demonstrate correct recognition and handling of merged requests at an endpoint
- Demonstrate correct recognition and handling of transport errors occurring during a transaction
- Demonstrate interoperable implementation of the entity type negotiation mechanisms (Accept, Accept-encoding, 415 Unsupported Media Type)
- Mechanisms for locating next destination
  - Demonstrate correct implementation of RFC3263 "locating a SIP server"
  - Demonstrate correct implementation of sending to a URI that utilizes explicit or default address, port, or transport attributes.
- Redirection mechanisms
  - Demonstrate successful redirection of a request (including having the redirected request succeed).
- Dialog mechanisms
  - Interoperable establishment and termination of a session-usage dialog
  - Interoperable establishment and termination of a session-usage dialog set up through a forking proxy.
  - Interoperable establishment and termination of multiple session-usages dialogs resulting from a single INVITE forking at a forking proxy.
  - Demonstrate ability to update a remote target using a re-INVITE
  - Demonstrate correct implementation of Route/Record-Route mechanism.
- Authentication mechanisms
  - Demonstrate successful authentication using SIP Digest
    - Correct implementation of 2069/2617 negotiation
    - Successful use of auth and auth-int using MD5
    - Correct implementation of rejecting unknown algorithms
    - Demonstrate successful authentication through multiple proxies, placed both in series and in parallel on different branches of a forked request
  - Demonstrate successful exchange of S/MIME protected entities
  - Demonstrate correct implementation of identifying a SIP peer using a TLS connection (mutual and server only auth)
- Extensibility mechanisms
  - Demonstrate correct implementation of rejecting unknown methods at endpoints
  - Demonstrate correct implementation of forwarding unknown methods at proxies
  - Demonstrate correct implementation of rejecting unknown
"Require"d extensions
- Demonstrate successful negotiation and exercise of some "Require"d extension
- Demonstrate correct implementation of ignoring unknown header fields at endpoints
- Demonstrate correct implementation of forwarding unknown header fields at proxies
- Demonstrate correct implementation of rejecting unknown SIP versions
- Demonstrate correct implementation of handling requests with unknown schemes in the RURI
- Demonstrate correct handling of unknown responses
- Demonstrate correct handling of bodies or body parts of unknown type
- Demonstrate correct implementation of ignoring unknown header field parameters at endpoints, and forwarding at proxies.

- URI specific features
  - Demonstrate successful transaction using a Request-URI containing escaped and non-ascii UTF8 characters
  - Demonstrate correct implementation of request generation based on a SIP URI containing embedded header fields
  - Demonstrate correct implementation of request generation based on a SIP URI containing an embedded entity (body)
  - Demonstrate correct implementation of SIP URI comparison
  - Demonstrate correct implementation of recommendations for constructing sip: URIs from tel: URIs

- Application-level features
  - Demonstrate successful offer/answer exchange using SIP
    - in INVITE/200
    - in 200/ACK
  - Demonstrate successful registration of an endpoint (creation, refreshing, and removal of a binding)
  - Demonstrate correct implementation of a "query" register
  - Demonstrate correct implementation of "soft-state" removal of unrefreshed bindings
  - Demonstrate correct manipulation of a single binding at an AoR with several bindings in place
  - Demonstrate correct implementation of wildcard de-registration
  - Demonstrate that someone will cut and paste this list into some inappropriate place without reading it before it's had review
  - Demonstrate successful query for capabilities (correct implementation of OPTIONS)

- Proxy application-level features
  - Demonstrate correct implementation of specified behavior upon receiving a CANCEL request
  - Demonstrate correct implementation of the Timer-C based mechanism for terminating an INVITE transaction
  - Demonstrate correct implementation of the Max-Forwards
mechanism
- Demonstrate correct implementation of loop-detection
  (what about the rest of fork-loop-fix)
- Demonstrate correct implementation of merging challenges
  received from multiple responses to a forked request
- Demonstrate correct implementation of merging multiple
  contacts received in redirect responses to a forked request
- Demonstrate correct implementation of recursing on
  redirections responses at proxies
- 2543-compatibility (that we might be able to drop?)
  - Successful exchange of messages without a Content-Length
    header over UDP
  - Successful transaction without a z9Gh4bK style transaction
    identifier
- Other unusual things (will we be able to find implementations?)
  - Demonstrate correct implementation of stateless UASs
  - Demonstrate correct implementation of stateless proxies
  - Demonstrate successfully setting an internal clock based on a
    Date header field returned in a REGISTER response.
  - Demonstrate successful discovery of a registrar through
    multicast to (224.0.1.72)

3. Open questions

Choosing the correct level of detail for the interoperability
statements in Section 2 is a balancing act. As
[I-D.dusseault-impl-reports] recommends, some of these statements
make other statements by implication. For instance, we are not (yet)
explicitly calling out testing case-sensitivity, multipart-mime,
provisional responses, or exercise of the registration duration
negotiation mechanisms. Instead, those are each already contained in
one or more of the statements made above and it is expected that the
report will call out any problems with these features at the
appropriate statement. Thus, we need to review the statements we
plan to make carefully to make sure they are fine enough to convince
ourselves that we have an opportunity to capture when things don't
work without becoming so fine that the sheer level of detail hides
the real statement of interoperability that we're trying to make. It
is certain that the list as presented so far is not complete.

Here are a few seed questions for exploring how well the statements
so far cover the specification (this is not intended to be
exhaustive):

- Right now there is nothing covering sips:. What do those
  statements need to look like?
What kind of statement will capture that we've ensured more than one implementation exercises 413 Request Entity too large and that clients do the right thing on receiving it?

Do we want to dive deeper into q-value processing at registrars and proxies?

How do we make a statement that captures we've found two implementations that discover their registrar through configuration rather than using URI resolution.

Do we dive any deeper than the statements above into proxy rewriting of Record-Route header field values in responses? (I propose no, that this a contained case as discussed above, but its less obvious that we'll be sure to remember to check and comment on problems if we can't find two independent implementations).

How do we make a statement that captures that we've verified independent implementation of the correct behavior for request timeouts at endpoints and proxies?

The list here _is_ intended to be mostly complete, however. Please review it carefully from that perspective and help identify what needs to be added and removed to take it all the way to complete.

4. Security Considerations

This document is not known to introduce any new security issues for consideration.

5. Informative References

[I-D.dusseault-impl-reports]
Dusseault, L. and R. Sparks, "Guidance on Interoperation and Implementation Reports",
draft-dusseault-impl-reports-00 (work in progress), July 2008.


Author's Address

Robert Sparks
Tekelec
17210 Campbell Road
Suite 250
Dallas, Texas 75252
USA

Email: RjS@nostrum.com