SIPPING WG
Internet-Draft
Expires: May 31, 2007

V. Gurbani, Ed.
Lucent Technologies/Bell Laboratories
C. Boulton
Ubiquity Software Corporation
R. Sparks
Estacado Systems
November 27, 2006

Session Initiation Protocol (SIP) Torture Test Messages for Internet Protocol Version 6 (IPv6)
draft-ietf-sipping-ipv6-torture-tests-00

Abstract

This informational document provides examples of Session Initiation Protocol (SIP) test messages designed to exercise and "torture" the IPv6 portions of a SIP implementation.

This work is being discussed on the sipping@ietf.org mailing list.

Table of Contents

1. Overview

This document is informational, and is NOT NORMATIVE on any aspect of SIP.

This document contains test messages based on the current version (2.0) of the Session Initiation Protocol as defined in [1].

This document is expected to be used as a companion document to the more general SIP torture test document [3], which does not include specific tests for IPv6 network identifiers.

This document does not attempt to catalog every way to make an invalid message, nor does it attempt to be comprehensive in exploring unusual, but valid, messages. Instead, it tries to focus on areas that may cause interoperability problems in IPv6 deployments.

The messages are presented in the text using a set of markup conventions to avoid ambiguity and meet Internet-Draft layout requirements. To resolve any remaining ambiguity, a bit-accurate version of each message is encapsulated in an appendix.

2. SIP and IPv6 Network Configuration

System-level issues like deploying a dual-stack proxy server, populating DNS with A and AAAA RRs, zero-configuration discovery of outbound proxies for IPv4 and IPv6 networks, when should a dual-stack proxy Record-Route itself, and media issues also play a major part in the transition to IPv6. This document does not, however, address these issues. Instead, a companion document [2] provides more guidance on these.

3. Parser Torture Tests

The test messages are organized into several sections. Some stress only a SIP parser and others stress both the parser and the application above it. Some messages are valid, and some are not. Each example clearly calls out what makes any invalid messages incorrect.

Please refer to the ABNF in [1] on representing IPv6 addresses in SIP. IPv6 addresses are delimited by a '[' and ']'.

The appendix contains an encoded binary form of all the messages and the algorithm needed to decode them into files.

3.1 Valid SIP request with raw IPv6 addresses

This REGISTER request is well-formatted per the grammar in [1]. An IPv6 address in presentation form appears in the Request-URI (R-URI), Via header, and Contact header.

Message Details: reg-good

```
REGISTER sip:[2001:db8::10] SIP/2.0
To: sip:user@example.com
From: sip:user@example.com;tag=81x2
Via: SIP/2.0/UDP [2001:db8::9:1];branch=z9hG4bKas3-111
Call-ID: SSG9559905523997077@hlau_4100
Contact: "Caller" <sip:caller@[2001:db8::1]>
CSeq: 98176 REGISTER
Content-Length: 0
```

3.2 Which port should I knock on?

IPv6 uses the colon to delimit octets. This may lead to ambiguity if the port number on which to contact a SIP server is inadverdently conflated with the IPv6 address. Consider the REGISTER request below. The sender of the request intended to specify a port number (5070). Unfortunately, however, since the IPv6 address in the R-URI is compressed, it makes it hard to tell whether the 5070 is a port number or the last octet in the address.

From a pure parsing point of view, the REGISTER request is well-formed. However, from a semantic point of view, it will not yield the desired result. Implementations must take care to ensure that when a raw IPv6 address appears in a SIP URI, then any port number must appear outside the closing '[' of the URI.

Message Details: reg-ambigous

```
REGISTER sip:[2001:db8::10:5070] SIP/2.0
To: sip:user@example.com
From: sip:user@example.com;tag=81x2
Via: SIP/2.0/UDP [2001:db8::9:1];branch=z9hG4bKas3-111
Call-ID: SSG9559905523997077@hlau_4100
Contact: "Caller" <sip:caller@[2001:db8::1]>
CSeq: 98176 REGISTER
Content-Length: 0
```

3.3 Knock on this port, please

In contrast to the example in Section 3.2, the following REGISTER request leaves no ambiguity whatsover on where the IPv6 address begins and where it ends. This REGISTER request is well formatted per the grammar in [1].

Message Details: reg-good-port

```
REGISTER sip:[2001:db8::10]:5070 SIP/2.0
To: sip:user@example.com
From: sip:user@example.com;tag=81x2
Via: SIP/2.0/UDP [2001:db8::9:1];branch=z9hG4bKas3-111
Call-ID: SSG9559905523997077@hlau_4100
Contact: "Caller" <sip:caller@[2001:db8::1]>
CSeq: 98176 REGISTER
Content-Length: 0
```

3.4 SIP request with IPv6 in Via received parameter

There currently exists an ambiguity on whether the received parameter of the Via header that contains an IPv6 address should have the delimiting '[' and ']' tokens. The RFC3261 ABNF indicates that this is not the case, however it makes the implementation of the parser more optimized if it was to recognize the '[' token as a beginning of an IPv6 address. In all the other instances where an IPv6 address is used in SIP, it is delimited by the '[' and ']' tokens. Thus, for the sake of orthogonality as well as optimized parsing, it seems appropriate that the IPv6 addresses in the received parameter be delimited by '[' and ']'. Some additional analysis on why the form that includes the delimiters is desirable is included in the following reference [7].

More specifically, RFC3261 ABNF defines the via-received production rule as follows:

via-received = "received" EQUAL (IPv4address / IPv6address)

IPv6address production rule is then defined to hold an IPv6 address without the delimiting '[' and ']' tokens. There is also an IPv6reference production rule in RFC3261 that yields the following:

IPv6reference = "[" IPv6address "]"

Thus, to allow the delimiting '[' and ']' tokens in the received

parameter, all that would need to be done is to amend the RFC3261 via-received production rule as follows:

via-received = "received" EQUAL (IPv4address / IPv6reference)

However, strong consensus has not yet emerged on this (the issue is documented on the SIPPING WG mailing list; see [6] for a link to the start of the discussion thread). At the 18th SIPit, it was observed that [5]:

> Those testing IPv6 made different assumptions about enclosing literal v6 addresses in Vias in []. By the end of the event, most implementations were accepting either. Its about 50/50 on what gets sent.

Consequently, as it now stands, implementations must follow the Robustness Principle [4] and be liberal in accepting a received parameter with or without the delimiting '[' and ']' tokens. When sending a request, implementations must not put the delimiting '[' and ']' tokens. The two test cases that follow, should thus be acceptable to any SIP implementation that supports IPv6.

3.4.1 SIP request with delimiting tokens in Via received parameter

This REGISTER request contains an IPv6 address in the Via received parameter. The IPv6 address is delimited by '[' and ']'. Even though this is not a well-formatted request based on a strict interpretation of the grammar in [1], robust implementations should nonetheless be able to parse the topmost Via header.

Message Details: reg-param

```
REGISTER sip:[2001:db8::10] SIP/2.0
To: sip:user@example.com
From: sip:user@example.com;tag=81x2
Via: SIP/2.0/UDP [2001:db8::9:1];received=[2001:db8::9:255];
   branch=z9hG4bKas3-111
Call-ID: SSG9559905523997077@hlau_4100
Contact: "Caller" <sip:caller@[2001:db8::1]>
CSeq: 98176 REGISTER
Content-Length: 0
```

3.4.2 SIP request without the delimiter tokens in the Via received parameter

This OPTIONS request contains an IPv6 address in the Via received paramter without the adorning '[' and ']'. This OPTIONS request is valid and well-formatted.

Message Details: opt-param

```
OPTIONS sip:[2001:db8::10] SIP/2.0
To: sip:user@example.com
From: sip:user@example.com;tag=81x2
Via: SIP/2.0/UDP [2001:db8::9:1];received=2001:db8::9:255;
   branch=z9hG4bKas3
Call-ID: SSG95523997077@hlau_4100
Contact: "Caller" <sip:caller@[2001:db8::1]>
CSeq: 921 OPTIONS
Content-Length: 0
```

3.5 SIP request with IPv6 identifiers in SDP body

This INVITE request is valid and well-formed. Notice the IPv6 addresses in the SDP body.

Message Details: inv-good

```
INVITE sip:user@[2001:db8::10] SIP/2.0
To: sip:user@[2001:db8::10]
From: sip:user@example.com;tag=81x2
Via: SIP/2.0/UDP [2001:db8::9:1];branch=z9hG4bKas3-111
Call-ID: SSG9559905523997077@hlau_4100
Contact: "Caller" <sip:caller@[2001:db8::1]>
CSeq: 8612 INVITE
Content-Type: application/sdp
Content-Length: 268

v=0
o=assistant 971731711378798081 0 IN IP6 2001:db8::20
s=Live video feed for today's meeting
c=IN IP6 2001:db8::1
t=3338481189 3370017201
m=audio 6000 RTP/AVP 2
a=rtpmap:2 G726-32/8000
m=video 6024 RTP/AVP 107
a=rtpmap:107 H263-1998/90000
```

3.6 Via headers from different networks in a request

This BYE request is valid and well-formed. The Via list contains a mix of IPv4 and IPv6 addresses.

Message Details: bye-good

```
BYE sip:user@host.example.com SIP/2.0
Via: SIP/2.0/UDP [2001:db8::9:1]:6050;branch=z9hG4bKas3-111
Via: SIP/2.0/UDP 192.0.2.1;branch=z9hG4bKjhja8781hjuaij65144
Via: SIP/2.0/TCP [2001:db8::9:255];branch=z9hG4bK451jj;
  received=192.0.2.200
Call-ID: 997077@lau_4100
CSeq: 89187 BYE
To: sip:user@example.net;tag=9817--94
From: sip:user@example.com;tag=81x2
```

3.7 SIP request with multiple network identifiers in SDP

This INVITE request is valid and well-formed. It contains multiple network identifiers in the SDP body.

Message Details: inv-mult-sdp

```
INVITE sip:user@[2001:db8::10] SIP/2.0
To: sip:user@[2001:db8::10]
From: sip:user@example.com;tag=81x2
Via: SIP/2.0/UDP [2001:db8::9:1];branch=z9hG4bKas3-111
Call-ID: SSG9559905523997077@hlau_4100
Contact: "Caller" <sip:caller@[2001:db8::1]>
CSeq: 8912 INVITE
Content-Type: application/sdp
Content-Length: 181

v=0
o=bob 280744730 28977631 IN IP4 host.example.com
s=
t=0 0
m=audio 22334 RTP/AVP 0
c=IN IP4 192.0.2.1
m=video 6024 RTP/AVP 107
c=IN IP6 2001:db8::1
a=rtpmap:107 H263-1998/90000
```

3.8 More test cases

TBD. Looking for more test cases...suggestions welcome.

4. Security Considerations

This document presents NON NORMATIVE examples of SIP session establishment. The security considerations in [1] apply.

Parsers must carefully consider edge conditions and malicious input as part of their design. Attacks on many Internet systems use crafted input to cause implementations to behave in undesirable ways. Many of the messages in this draft are designed to stress a parser implementation at points traditionally used for such attacks. This document does not, however, attempt to be comprehensive. It contains some common pitfalls that the authors have discovered while parsing IPv6 identifiers in SIP implementations.

5. IANA Considerations

This document has no actions for IANA.

6. Acknowledgments

The authors acknowledge Jeroen van Bemmel, Dennis Bijwaard, Gonzalo Camarillo, Bob Gilligan, Larry Kollasch, Erik Nordmark, Kumiko Ono and Pekka Pessi for input provided during the construction of the document and discussion of the test cases.

The appendix contains a bit-exact archive of each message following the convention established by Robert Sparks.

7. References

7.1 Normative References

[1] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[2] Camarillo, G., El Malki, K., and V. Gurbani, "IPv6 Transition in the Session Initiation Protocol (SIP)", draft-ietf-sipping-v6-transition-02.txt (work in progress), October 2005.

7.2 Informative References

[3] Sparks, R., Hawrylyshen, A., Hawrylyshen, A., Rosenberg, J., and H. Schulzrinne, "Session Initiation Protocol Torture Test Messages", draft-ietf-sipping-torture-tests-09 (work in progress), November 2005.

[4] Braden, R., "Requirements for Internet Hosts -- Communication Layers", RFC 1122, October 1989.

[5] Sparks, R., "preliminary report: SIPit 18", Electronic Mail archived at http://www1.ietf.org/mail-archive/web/sip/current/msg14103.html, April 2006.

[6] Gurbani, V., "SIP/IPv6 torture test and possible interaction with rfc3261 ABNF", Electronic Mail archived at http://www1.ietf.org/mail-archive/web/sipping/current/msg10341.html, February 2006.

[7] van Bemmel, J., "[Sipping] Re: [Sip-implementors] SIP/IPv6 torture test and possible interaction with rfc3261 ABNF", Electronic Mail archived at http://www1.ietf.org/mail-archive/web/sipping/current/msg10373.html, February 2006.

Authors' Addresses

Vijay Gurbani (editor)
Lucent Technologies/Bell Laboratories
2701 Lucent Lane
Rm 9F-546
Lisle, IL 60532
USA

Phone: +1 630 224 0216
Email: vkg@lucent.com

Chris Boulton
Ubiquity Software Corporation
Building 3
West Fawr Lane
St Mellons
Cardiff, South Wales CF3 5EA

Email: cboulton@ubiquitysoftware.com

Robert J. Sparks
Estacado Systems

Email: RjS@estacado.net

Appendix A. Bit-exact archive of each test message

The following text block is an encoded, gzip compressed TAR archive of files that represent each of the example messages discussed in Section 4.

To recover the compressed archive file intact, the text of this document may be passed as input to the following Perl script (the output should be redirected to a file or piped to "tar -xzvf -").

```
#!/usr/bin/perl
use strict;
my $bdata = "";
use MIME::Base64;
while(<>) {
  if (/-- BEGIN MESSAGE ARCHIVE --/ .. /-- END MESSAGE ARCHIVE --/) {
        if ( m/^\s*[^\s]+\s*$/) {
            $bdata = $bdata . $_;
        }
   }
}
print decode_base64($bdata);
```

Alternatively, the base-64 encoded block can be edited by hand to remove document structure lines and fed as input to any base-64 decoding utility.

A.1 Encoded Reference Messages

```
-- BEGIN MESSAGE ARCHIVE --
H4sICEzabEQAA2RhdGEudGFyAO2Z72/iNhjH+zp/hXVv9irgxz/jdJlu63Ud2ukOFVZpm
qrJgA/CCMmSgK776+eQBgq0pTs1VGvzFRBCbOzYz+frHxncGHccx6OTGoUBY8HYif3ElI
k7x1KcihPAlABhggK1vwMhgE9wnZWqtMhyndoil3+NH0136HpxI5SQ4igxEc9ZxTr10+/
nKAsTf5GZ9P0kzvKW+aqjZGZawzhCvU63TVrYuQq1X520f/vQRX8Q20n+aOD5vvLh2heY
49NBqufDSfCPmlywwa86oy4A7OcF3LIv0qI7GaaTqfakB5PpQodTwYGx7cz9s52CCefXO
3/COEynp6kZmnBpRkFVFhDsnOnZzO188JFSEkv5fqYXfzIbms5Zz/ztI0+BJ5FtD6cf+5
s2qZpjbvLTXI8D5YF0XcWcn9M4uiedbbZVOg++Euele/ewwvnypfkHGzsl/5jZNy/5x6L
h/wjqfLrq9O9YwB2+AF+vDWALie00TwLhkIE84B1rZHu9C8W5UphzQm/5nWwAjue5HuY+
eldkMOk79H1RneHqZKu61z9UtAsgqLz5VXYzz93+TWJ8pJNkFg51HsbzdjZK1lc/mvk4n
/iICM9xlgF24kBnWWjDZ54jJUFSkABUelJ52AOE7d+jTlegTfHWhLLgozUmtAxHJkZfjB
mhL3GK8nikb77LUGRMHs7HzjDYywtOHlBKPeYBeApRKu0lSTA4UaAXozBGwoYeuux32z9
edRFxdJDmSaQTn6ALSYRLSduzKWzysnCBCVsnByw3GewJ+oUI2wVKeW1VRPb/wMkafYsK
/48Ws9y1oV5XGYfmf5Kz0v85I5hD4f/UDgWN/x9Bb9b/1bf6P3hQ+f8gHiDiYcmYpNh+U
1LaBUzp+wztzqat91sPxwivHZsQSjcejCvbZwiUbadi2vqwW987RDxm4eheD4+T3E10qq
MaY+wQ/4wV6z9GKXCOoZz/sWIeWGOd1nrj/H/u9jufP/VW+D6F/bvx/Czgr5dqOwu7Uwe
ttGcMu6bwPH6gCKDbttgj/jXPflIzdnU0CMeLeJHVVMZB/ims13+43P+hhDbj/zF0eX7R
6fXPL/cNwOf2Ro7lAsce/otdHIGqm7+H+FeM/JYK/l96/5cRVvFvewXf7v9Aw/8R9DD/D
fpvQRX/bhKneU1l/Kfxn7LV+A+cNfwfQY/wv5oANCbwulXw/+Lrf7F5/oPFav+PUNbwfw
zVPv6jp+8AHHq229hCo0aNGj2f/gWwk3L/ACYAAA==
-- END MESSAGE ARCHIVE --
```

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.