SIP working group IETF#70
Essential corrections

Keith Drage

Essential corrections

- draft-drage-sip-essential-correction-02
- An essential change is one where in the absence of the correction, it will not be possible to implement the specification contained in the original RFC in a manner to ensure interoperability or correct operation.
- See: http://www.softarmor.com/mediawiki/index.php/Essential_Corrections_Tracking

Format (1)

- In addition to the normal rules for contents of a standards track RFC, sections to the RFC should document the following (probably as separate sections or subsections):
- Reason for change. Text which explains why the change is necessary. This should be focussed on identifying why the text in the existing RFC is incorrect.
- Summary of change. Enter text which describes the most important components of the change. i.e. how the change is made.
- Consequences if not approved. Enter here the consequences if this change were to be rejected. Explain the issues that implementations will have in the absence of this change, i.e. what fails to operate correctly. This text should be drafted such that the working group can make a decision as to whether the change is essential or not.

Format (2)

- The change. Provide only the normative changes outside the context of the sections of the corrected RFC. This section is for those implementors who want to understand the normative changes at an immediate view.
- OPEN ISSUE: The above element has been inserted at the request of participants at IETF#69. The above element requires further study, both in the format it should take, and what occurs if after publication, it is found to differ from the next element. Should one element take precedence over the other, or do we sort it out at the next reissue of the change RFC.
- The change in detail. Clearly identify the section of the RFC to be changed, and show precisely how the text changes. An implementor should be able to take the original RFC and edit the change as described to obtain the new approved text.

Errata

- Do not make normative changes to the specification and have been underused in the SIP specifications.

draft-ietf-sip-record-route-fix-01

A typical function of a Session Initiation Protocol (SIP) Proxy is to set a Record-Route header on initial requests in order to make subsequent requests pass through it. This header contains a SIP Uniform Resource Identifier (URI) indicating where and how the subsequent requests should be sent to reach the proxy. Like any SIP URI, it can contain sip or sips schemes, IPV4 or IPV6 addresses, and URI parameters that could influence the routing like different transport parameters (UDP, TCP, SCTP...), or a compression indication like "comp=sigcomp". When a proxy has to change some of those parameters between its incoming and outgoing interfaces (multi-homed proxies, transport protocol switching, sip to sips or IPV4 to IPV6 scenarios...), the question arises on what should be put in Record-Route header(s). It is just not possible to make one header having the characteristics of both sides at the same time. This document aims to clarify these scenarios and fix bugs already identified on this topic; it formally recommends the use of the double Record-Route technique as an alternative to the current RFC3261 text, which only describes Record-Route rewriting solution.

draft-gurbani-sip-ipv6-abnf-fix-00

- This memo corrects the Augmented Backus-Naur Form (ABNF) production rule associated with generating IPv6 literals in RFC3261

draft-hilt-sip-correction-503-01

- Overload occurs in the Session Initiation Protocol (SIP) when SIP servers have insufficient resources to process all SIP messages they receive. The SIP protocol specified in RFC 3261 provides the 503 (Service Unavailable) response code as a remedy for servers under overload. However, the current definition of 503 (Service Unavailable) has problems and can in fact amplify an overload condition. This document proposes an essential correction to RFC
- Defines two options – we need to choose one to proceed

Option 1

1. Introduce a new response code, 507 (Server Overload), for servers temporarily unavailable due to overload. This response is similar to a 500 (Server Internal Error) response. Its Retry-After header has the same semantics (i.e., it only affects the current request) and it is forwarded all the way to the UAC.
2. A difference between a 500 (Server Internal Error) and a 507 (Server Overload) response is that a 507 (Server Overload) response should not be re-tried at an alternate server. Instead, it should be returned to the UAC. This way, excess requests are quickly cleared from a network of SIP servers. A new header, "Allow-Retry", may be used to explicitly allow proxies to re-try the request at an alternate server.
3. Deprecate the use of 503 (Service Unavailable) responses for temporary unavailability due to overload.
4. Change dropping requests or refusing the connection as a replacement for sending a 503 (Service Unavailable) response from MAY to SHOULD NOT.
5. Recommend the use of IP addresses for blocking traffic after receiving a 503 (Service Unavailable) with Retry-After and not the hostname.

Option 2

1. Deprecate the use of Retry-After headers in 503 (Service Unavailable) responses for overload control by servers with a small client population (< 20 clients). The use of Retry-After remains unchanged for servers with a large number of clients such as edge proxies (> 20 clients) and server maintenance. Proxies that create a 500 (Server Internal Error) response after receiving a 503 (Service Unavailable) may include a Retry-After header in the 500 (Server Internal Error) response to prevent the UAC from instantly retrying the request.
2. Introduce a new header, "Allow-Retry", for 503 (Service Unavailable) responses. This header controls whether a client receiving a 503 (Service Unavailable) response should or should not forward the request to an alternate server. The default value for this header is true. A somewhat simplistic alternative to the introduction of a new header is to deprecate forwarding requests to alternate servers if the 503 (Service Unavailable) response does not contain a Retry-After header. In this case, it can be assumed that it was created because of overload and not server maintenance.
3. Change dropping requests or refusing the connection as a replacement for sending a 503 (Service Unavailable) response from MAY to SHOULD NOT.
4. Recommend the use of IP addresses for blocking traffic after receiving a 503 (Service Unavailable) with Retry-After and not the hostname.

draft-dotson-sip-mutual-auth-00

- This document defines updates to the Session Initiation Protocol (SIP) to add mutual authentication to proxy authentication. The Proxy-Authentication-Info header, which allows a UA to authenticate a proxy when challenged, is not defined in SIP ([RFC 3261]). Supporting mutual proxy authentication in SIP would mitigate certain risks in using SIP Digest proxy authentication.

draft-sparks-sip-invfix-00

- This document normatively updates RFC 3261, the Session Initiation Protocol (SIP), to address an error in the specified handling of success (200 class) responses to INVITE requests. Elements following RFC 3261 exactly will misidentify retransmissions of the request as a new, unassociated, request. The correction involves modifying the INVITE transaction state machines and changing the way responses that