A Common Log Format for SIP using IPFIX Files

draft-niccolini-sipclf-ipfix-05

Saverio Niccolini Benoit Claise *Brian Trammell*
Hadriel Kaplan

IETF 79, November 6-13, 2010 Beijing, China

Outline

The IPFIX Approach to SIPCLF

Changes since Maastricht

Next Steps

IPFIX Files provide a basis for flexible logging

- Efficient, self-describing framing based on templates
- Optimized for fast export/storage of high-volume, relatively semantically uncomplicated data
    - More complicated semantics supported by draft-ietf-ipfix-structured-data
- Information elements provided for binary-representation of common network-related data (e.g. IP addresses, timestamps)
    - Many of these are applicable to and reused by SIPCLF

Definition of new SIP-specific Information Elements

- sipMethod: method encoded as 8-bit integer, by order in SIP Parameters Methods registry
- sipResponseStatus: presence signifies a response record
- sipObservationType: what role did the observer have in the message?
- URIs: sipRequestURI, sipFromURI, sipFromTag, sipToURI, sipToTag, sipContactURI, sipPaiURI
- Identifiers: sipCallId, sipSessionId, sipSequenceNumber, sipAuthUsername
- Message dump support: sipMessageSection, sipMessageSectionOffset, sipMessageLength

Definition an IPFIX SIPCLF log file

- Draft defines base templates for SIP Request and SIP Response log entry
  - Additional templates for e.g. IPv4 vs. IPv6 endpoints, optional records
  - Supplemental optional templates for raw SIP message dumping
- A SIPCLF log file is then simply any IPFIX File containing Templates based on these base Templates, and records defined by them
  - May contain additional information elements in SIPCLF templates (e.g. optional additional data, vendor-specific features)
  - May contain data described by non-SIPCLF templates (e.g. multi-application logging, combination with flows from data plane)

More examples

- Examples now provided for each example in the Problem Statement
  - Generated by a running implementation based on ripfix
- Examples for torture tests *not* updated since decision that logging should preserve SIP escaping intact
  - Torture tests mainly test the SIP parser in front of logging
  - Length prefix encoding for IPFIX strings mitigates string handling danger
  - Still not clear that these are useful in this document (open issue)

No more bodies, but raw messages

- Per list discussion, body logging not in scope for SIPCLF
  - debug dumps and logs are separate things
- Body logging mechanism repurposed as *raw message* logging mechanism
  - Minimum handling by logging process: what you saw is what you get
  - Still provides logging of large raw Messages in multiple IPFIX messages

Lots of discussion

- Not really a change, but led to delays on the original "choose one and finish specifying it for Beijing" target.
- sipMethod: text (to support weird methods) or *integer* (registered methods only)?
- endpoint logging: FQDNs or *IP addresses* mandatory?

Open issues

- New information elements defined in PEN 35566 (trammell.ch) space to facilitate early implementation testing.
    - Assign these real numbers from IANA on WG adoption.
- Peter Musgrave implemented both side-by-side to test efficiency (thanks, Peter!); results should be incorporated in next revision.
- Torture tests should be updated or removed, as necessary.
- Mechanism for cross-referencing multiple records related to the same SIP message (e.g., large raw message logging), if necessary.

SIPCLF WG item?

- Specification relatively mature
    - Two implementations (based on same IPFIX core)
- If selected, completion in Prague timeframe possible

Questions, comments, flames?