Session Initiation Protocol (SIP) Common Log Format (CLF)
(draft-gurbani-sipclf-problem-statement-00)

Editors:
Vijay K. Gurbani <vkg@bell-labs.com> Bell Laboratories/Alcatel-Lucent
Eric Burger <eburger@standardstrack.com> Neustar, Inc.
Contributors

Humberto Abdelnur <Humberto.Abdelnur@loria.fr>
Tricha Anjali <tricha@ece.iit.edu>
Oliver Festor <Olivier.Festor@loria.fr>
Hadriel Kaplan <hkaplan@acmepacket.com>
Adam Roach <adam@nostrum.com>
Theo Zourzouvillys <theo@voip.co.uk>
Dale Worley <dworley@nortel.com>
What is CLF

Common Log Format (CLF):
A summary of an application layer PDU*

* (To paraphrase RjS)
Need for a CLF

SIP has many entities participating in a session setup request. Need some way to find out what is going on in real-time or post process. Model: HTTP CLF!
HTTP CLF


- **IP address of client**: Making the request
- **Remote logname of user as determined by rfc931**: User identification
- **Date/time access was made**: Timestamp
- **Request line**: GET /apache_pb.gif HTTP/1.0
- **Status code (response returned by server)**: 200 (OK)
- **Length of document transferred**: 2326 bytes
What SIP CLF is and is not ...

SIP CLF is NOT...

- ... a replacement for a CDR (Call Detail Record).
- ... a billing tool.
- ... a QoS measurement tool.

SIP CLF IS:

- ... a standardized format that can be used by all SIP entities.
- ... an easily digestible log of past and current transactions.
- ... a format that allows quick parsing to discover relationship between transactions
  
  $ grep yuhyt6 sip-clf.txt
  
  gets all transactions with this label.
- ... amenable for easy parsing and creating other innovative tools.
Use cases

• Trend analysis ("I want to find out which geographical area are the most calls coming from at 2:00 AM").
• Troubleshooting ("How long did it take to generate a final response to an INVITE?")
• Message correlation across transactions ("Find all messages corresponding to Call-ID X, including all forked branches")
• Transaction correlation across dialogs ("Find all messages for dialog created by Call-ID X and tags A and B").
• Establish concise and standardized diagnostic trail of a SIP session locally and globally.
• Establish concise and standardized format for training automata (anomaly detection.)
Benefits of a SIP CLF

• Establishes a common reference for logging SIP messages across vendor/open-source implementations.
• Correlate SIP messages across transactions and dialogs.
• Easily search, merge, and summarize log records.
• Train anomaly detection systems to trigger alarms.
• Allow independent tool providers to provide innovative tools for trend analysis and traffic reports.
• Common diagnostic trail from testing of SIP equipment.
• Can be used for off-line analysis (trend analysis) as well as real-time analysis.
Challenges in defining SIP CLF

- SIP is not a *linear* request-reply protocol
  - HTTP is *linear*: pipelining okay, one request = one response.

- Complexity inherent in the protocol:
  - Serial and parallel forking elicit multiple responses.
  - Delays between getting a request and sending a response (outside of “long polling” in HTTP, servers respond quickly; not quite so in SIP. Impact on proxies.)
  - Multiple transactions grouped in a dialog; dialog persists for a long time, transactions short-lived (e.g., BYE comes much later, but relation between INVITE and BYE should be preserved in a log file.)
Challenges in defining SIP CLF

- ACK requests need careful considerations:
  - Only tied to an INVITE.
  - No responses for ACKs.
  - For non-2xx, ACKs hop-by-hop (part of INVITE transaction.)
  - For 2xx, ACK end-to-end.

- CANCEL requests need careful considerations:
  - Only tied to an INVITE.
  - Requires exactly one response.
  - Is propagated hop-by-hop.
Challenges in defining SIP CLF

• INVITE can pend, resulting in a 1xx response (200ms rule.) This 1xx response needs to be captured to train automata.

• SIP has a richer set of actors: UAS, UAC, B2BUA, proxy, registrar, redirect server, ...

• Need to take SIP extensibility in account.

• Preserve user privacy in CLF (through anonymization, etc.)
SIP CLF fields

Date
Remotehost
Authuser
Method
Request-URI
From (including tag)
To (including tag)
Call-ID
Status
Contact list
Server-transaction
Client-transaction
For sake of illustration only, example is in ASCII:

- A proxy receives a request and sends a provisional upstream

```<allOneLine>
  1230756560 192.168.1.10 - INVITE sip:bob@example.net
  sip:alice@example.com;tag=hy7 sip:bob@example.net
  7yhgt1@example.com - uyt67h FORK/-
</allOneLine>
  1230756560 uyt67h - 100 INVITE + -```
SIP CLF example

The proxy forks two branches:

<allOneLine>
  1230756563 - - INVITE sip:bob@home.example.net
  sip:alice@example.com;tag=hy7 sip:bob@example.net
  7yhgt1@example.com - uyt67h CLIENT/hb76
</allOneLine>
<br>
<allOneLine>
  1230756564 - - INVITE sip:bob@carphone.example.net
  sip:alice@example.com;tag=hy7 sip:bob@example.net
  7yhgt1@example.com - uyt67h CLIENT/hb77
</allOneLine>
SIP CLF example

Proxy receives provisionals and final responses:

```
1230756565 uyt67h hb76 100 INVITE sip:bob@example.net;tag=876v -
1230756565 uyt67h hb77 100 INVITE sip:bob@example.net;tag=561t -
1230756565 uyt67h hb76 180 INVITE sip:bob@example.net;tag=876v -
1230756565 uyt67h hb77 180 INVITE sip:bob@example.net;tag=561t -
1230756567 uyt67h hb77 182 INVITE sip:bob@example.net;tag=561t -
1230756568 uyt67h hb76 500 INVITE sip:bob@example.net;tag=876v -
```

```
1230756568 uyt67h hb77 200 INVITE sip:bob@example.net;tag=561t "sip:bob@home.example.net"
```

```
SIP CLF example

Proxy sends 200 OK upstream and ACKs 500:

<allOneLine>
  1230756569 uyt67h - 200 INVITE
  sip:bob@example.net;tag=561t
  "sip:bob@home.example.net"
</allOneLine>
<allOneLine>
  1230756569 + - ACK sip:bob@home.example.net + + + -
  uyt67h CLIENT/hb76
</allOneLine>
Solutions space

- PCAP representation (not actively pursued by author; see http://tools.ietf.org/html/draft-kaplan-sipping-clf-pcap-00)
- IPFIX representation (no drafts exist yet.)