The Common Log Format (CLF) for the Session Initiation Protocol (SIP)
draft-gurbani-sipclf-problem-statement-01

Abstract

Well-known web servers such as Apache and web proxies like Squid support event logging using a common log format. The logs produced using these de-facto standard formats are invaluable to system administrators for trouble-shooting a server and tool writers to craft tools that mine the log files and produce reports and trends. Furthermore, these log files can also be used to train anomaly detection systems and feed events into a security event management system. The Session Initiation Protocol does not have a common log format, and as a result, each server supports a distinct log format that makes it unnecessarily complex to produce tools to do trend analysis and security detection. We propose a common log file format for SIP servers that can be used uniformly by proxies, registrars, redirect servers as well as back-to-back user agents.

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 July 19, 2010.

Copyright Notice

Copyright (c) 2010 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 (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the BSD License.
Table of Contents

1. Terminology .................................................. 3
2. Introduction .................................................. 3
3. Problem statement ............................................. 4
4. What SIP CLF is and what it is not .......................... 4
5. Alternative approaches to SIP CLF ......................... 5
   5.1. SIP CLF and CDRs ......................................... 5
   5.2. SIP CLF and Wireshark packet capture .................. 6
6. Motivation and use cases ..................................... 6
7. Challenges in establishing a SIP CLF ....................... 8
8. SIP CLF fields ............................................... 9
9. Security Considerations ...................................... 11
10. Operational guidance ......................................... 12
11. IANA Considerations ......................................... 12
12. Acknowledgments ............................................. 12
13. References .................................................. 12
   13.1. Normative References ................................. 12
   13.2. Informative References ............................... 12
Authors’ Addresses ............................................. 13
1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [1].

RFC 3261 [2] defines additional terms used in this document that are specific to the SIP domain such as "proxy"; "registrar"; "redirect server"; "user agent server" or "UAS"; "user agent client" or "UAC"; "back-to-back user agent" or "B2BUA"; "dialog"; "transaction"; "server transaction".

This document uses the term "SIP Server" that is defined to include the following SIP entities: user agent server, registrar, redirect server, a SIP proxy in the role of user agent server, and a B2BUA in the role of a user agent server.

2. Introduction

Servers executing on Internet hosts produce log records as part of their normal operations. A log record is, in essence, a summary of an application layer protocol data unit (PDU), that captures in precise terms an event that was processed by the server. These log records serve many purposes, including analysis and troubleshooting.

Well-known web servers such as Apache and Squid support event logging using a Common Log Format (CLF), the common structure for logging requests and responses serviced by the web server. It can be argued that a good part of the success of Apache has been its CLF because it allowed third parties to produce tools that analyzed the data and generated traffic reports and trends. The Apache CLF has been so successful that not only did it become the de-facto standard in producing logging data for web servers, but also many commercial web servers can be configured to produce logs in this format. An example of Apache CLF is depicted next:

```
%h %l %u %t "%r" %s %b
remotehost rfc931 authuser [date] request status bytes
```

remotehost: Remote hostname (or IP number if DNS hostname is not available, or if DNSLookup is Off.

rfc931: The remote logname of the user.
authuser:  The username by which the user has authenticated himself.

[date]:  Date and time of the request.

request:  The request line exactly as it came from the client.

status:  The HTTP status code returned to the client.

bytes:  The content-length of the document transferred.

3. Problem statement

The Session Initiation Protocol [2](SIP) is an Internet multimedia session signaling protocol that is increasingly used for other services besides session establishment. A typical deployment of SIP in an enterprise will consist of SIP entities from multiple vendors. Currently, if these entities are capable of producing a log file of the transactions being handled by them, the log files are produced in a proprietary format. The result of multiplicity of the log file formats is the inability of the support staff to easily trace a call from one entity to another, or even to craft common tools that will perform trend analysis, debugging and troubleshooting problems uniformly across the SIP entities of multiple vendors.

SIP does not currently have a CLF format and this document serves to provide the rationale to establish a SIP CLF and identifies the required minimal information that must appear in any SIP CLF record.

4. What SIP CLF is and what it is not

The SIP CLF is a standardized manner of producing a text file. This format can be used by SIP clients, SIP Servers, proxies, and B2BUAs. The SIP CLF is simply an easily digestible log of currently occurring events and past transactions. It contains enough information to allow humans and automata to derive relationships between discrete transactions handled at a SIP entity. For example, a SIP administrator should be able to issue a concise command to discover relationships between transactions or to search a certain dialog or transaction.

Note: The exact form of the "concise command" is left unspecified until the working group agrees to one or more formats for encoding the fields.
The SIP CLF is amenable to quick parsing (i.e., well-delimited fields) and it is platform and operating system neutral.

The SIP CLF is amenable to easy parsing and lends itself well to creating other innovative tools.

The SIP CLF is not a billing tool. It is not expected that enterprises will bill customers based on SIP CLF. The SIP CLF records events at the signaling layer only and does not attempt to correlate the veracity of these events with the media layer. Thus, it cannot be used to trigger customer billing.

The SIP CLF is not a quality of service (QoS) measurement tool. If QoS is defined as measuring the mean opinion score (MOS) of the received media, then SIP CLF does not aid in this task since it does not summarize events at the media layer.

5. Alternative approaches to SIP CLF

It is perhaps tempting to consider other approaches --- which though not standardized, are in wide enough use in networks today --- to determine whether or not a SIP CLF would benefit a SIP network consisting of mutil-vendor products. The two existing approaches that approximate what SIP CLF does are Call Detail Records (CDRs) and Wireshark packet sniffing.

5.1. SIP CLF and CDRs

CDRs are used in operator networks widely and with the adoption of SIP, standardization bodies such as 3GPP have subsequently defined SIP-related CDRs as well. Today, CDRs are used to implement the functionality approximated by SIP CLF, however, there are important differences.

One, SIP CLF operates natively at the transaction layer and maintains enough information in the information elements being logged that dialog-related data can be subsequently derived from the transaction logs. Thus, esoteric SIP fields and parameters like the To header, including tags; the From header, including tags, the CSeq number, etc. are logged in SIP CLF. By contrast, a CDR is used mostly for charging and thus saves information to facilitate that very aspect. A CDR will most certainly log the public user identification of a party requesting a service (which may not correspond to the From header) and the public user identification of the party called party (which may not correspond to the To header.) Furthermore, the sequence numbers maintained by the CDR may not correspond to the SIP CSeq header. Thus it will be hard to piece together the state of a
dialog through a sequence of CDR records.

Two, a CDR record will, in all probability, be generated at a SIP entity performing some form of proxy-like functionality of a B2BUA providing some service. By contrast, SIP CLF is light-weight enough that it can be generated by a canonical SIP user agent server and user agent client as well, including those that execute on resource constrained devices (mobile phones).

Finally, SIP is also being deployed outside of operator-managed VoIP networks. Universities, research laboratories, and small-to-medium size companies are deploying SIP-based VoIP solutions on networks owned and managed by them. Much of the latter constituencies will not have an interest in generating CDRs, but they will like to have a concise representation of the messages being handled by the SIP entities in a common format.

5.2. SIP CLF and Wireshark packet capture

Wireshark is a popular raw packet capture tool. It contains filters that can understand SIP at the protocol level and break down a captured message into its individual header components. While Wireshark is appropriate to capture and view discrete SIP messages, it does not suffice to serve in the same capacity as SIP CLF for two reasons.

First, while the Wireshark format saves bulk of the information needed to create transaction and dialog state, the Wireshark format is a binary format that does not lend itself very well to being manipulated by text-based tools. Second and more importantly, if the SIP messages are exchanged over a TLS-oriented transport, Wireshark will be unable to decrypt them and render them as individual SIP headers.

6. Motivation and use cases

As SIP becomes pervasive in multiple business domains and ubiquitous in academic and research environments, it is beneficial to establish a CLF for the following reasons:

Common reference for interpreting events: In a laboratory environment or an enterprise service offering there will typically be SIP servers from multiple vendors participating in routing requests. Absent a CLF format, each server will produce output records in a native format making it hard to establish commonality for tools that operate on the log file.
Writing common tools: A CLF format allows independent tool providers to craft tools and applications that interpret the CLF data to produce insightful trend analysis and detailed traffic reports. The format should be such that it retains the ability to be read by humans and processed using traditional Unix text processing tools.

Session correlation across diverse processing elements: In operational SIP networks, a request will typically be processed by more than one SIP server. A SIP CLF will allow the network operator to trace the progression of the request (or a set of requests) as they traverse through the different servers to establish a concise diagnostic trail of a SIP session.

Note that tracing the request through a set of servers is considerably less challenging if all the servers belong to the same administrative domain.

Message correlation across transactions: A SIP CLF can enable a quick lookup of all messages that comprise a transaction (e.g., "Find all messages corresponding to server transaction X, including all forked branches."

Message correlation across dialogs: A SIP CLF can correlate transactions that comprise a dialog (e.g., "Find all messages for dialog created by Call-ID C, From tag F and To tag T.")

Trend analysis: A SIP CLF allows an administrator to collect data and spot patterns or trends in the information (e.g., "What is the domain where the most sessions are routed to between 9:00 AM and 12:00 PM?")

Train anomaly detection systems: A SIP CLF will allow for the training of anomaly detection systems that once trained can monitor the CLF file to trigger an alarm on the subsequent deviations from accepted patterns in the data set. Currently, anomaly detection systems monitor the network and parse raw packets that comprise a SIP message -- a process that is unsuitable for anomaly detection systems [3]. With all the necessary event data at their disposal, network operations managers and information technology operation managers are in a much better position to correlate, aggregate, and prioritize log data to maintain situational awareness.
Testing: A SIP CLF allows for automatic testing of SIP equipment by writing tools that can parse a SIP CLF file to ensure behavior of a device under test.

Troubleshooting: A SIP CLF can enable cursory trouble shooting of a SIP server (e.g., "How long did it take to generate a final response for the INVITE associated with Call-ID X?")

Offline analysis: A SIP CLF allows for offline analysis of the data gathered. Once a SIP CLF file has been generated, it can be transported (subject to the security considerations in Section 9) to a host with appropriate computing resources to perform subsequent analysis.

Real-time monitoring: A SIP CLF allows administrators to visually notice the events occurring at a SIP server in real-time providing accurate situational awareness.

7. Challenges in establishing a SIP CLF

Establishing a CLF for SIP is a challenging task. The behavior of a SIP entity is more complex when compared to the equivalent HTTP entity.

Base protocol services such as parallel or serial forking elicit multiple final responses. Ensuing delays between sending a request and receiving a final response all add complexity when considering what fields should comprise a CLF and in what manner. Furthermore, unlike HTTP, SIP groups multiple discrete transactions into a dialog, and these transactions may arrive at a varying inter-arrival rate at a proxy. For example, the BYE transaction usually arrives much after the corresponding INVITE transaction was received, serviced and expunged from the transaction list. Nonetheless, it is advantageous to relate these transactions such that automata or a human monitoring the log file can construct a set consisting of related transactions.

ACK requests in SIP need careful consideration as well. In SIP, an ACK is a special method that is associated with an INVITE only. It does not require a response, and furthermore, if it is acknowledging a non-2xx response, then the ACK is considered part of the original INVITE transaction. If it is acknowledging a 2xx-class response, then the ACK is a separate transaction consisting of a request only (i.e., there is not a response for an ACK request.) CANCEL is another method that is tied to an INVITE transaction, but unlike ACK, the CANCEL request elicits a final response.
While most requests elicit a response immediately, the INVITE request in SIP can pend at a proxy as it forks branches downstream or at a user agent server while it alerts the user. RFC 3261 [2] instructs the server transaction to send a 1xx-class provisional response if a final response is delayed for more than 200 ms. A SIP CLF log file needs to include such provisional responses because they help train automata associated with anomaly detection systems and provide some positive feedback for a human observer monitoring the log file.

Finally, beyond supporting native SIP actors such as proxies, registrars, redirect servers, and user agent servers (UAS), it is beneficial to derive a CLF format that supports back-to-back user agent (B2BUA) behavior, which may vary considerably depending on the specific nature of the B2BUA.

8. SIP CLF fields

The inspiration for the SIP CLF is the Apache CLF. However, the state machinery for a HTTP transaction is much simpler than that of the SIP transaction (as evidenced in Section 7). The SIP CLF needs to do considerably more.

Accordingly, the following SIP CLF fields are defined as minimal information that must appear in any SIP CLF record:

- **date**: Date and time of the request or response represented as the number of seconds and milliseconds since the Unix epoch.
- **remotehost**: The DNS name or IP address of the upstream client.
- **authuser**: The user name by which the user has been authenticated. If the user name is unknown or when a request is challenged, the value in this field must be "-"
- **method**: The upper-case name of the SIP method.
- **request-uri**: The Request-URI, including any URI parameters.
- **from**: The From URI, including the tag. Whilst one may question the value of the From URI in light of RFC4744 [4], the From URI, nonetheless, imparts some information. For one, the From tag is important and, in the case of a REGISTER request, the From URI can provide information on whether this was a third-party registration or a first-party one.
to: The To URI, including tag.

callid: The Call-ID.

status: The SIP response status code returned upstream.

contactlist: Contact URIs in the response, if any. A "-" field value may be used if there aren’t any Contact URIs.

server-txn: Server transaction identification code - the transaction identifier associated with the server transaction. Implementations can reuse the server transaction identifier (the topmost branch-id of the incoming request, with or without the magic cookie), or they could generate a unique identification string for a server transaction (this identifier needs to be locally unique to the server only.) This identifier is used to correlate ACKs and CANCELs to an INVITE transaction; it is also used to aid in forking as explained later in this section.

client-txn: Client transaction identification code - this field is used to associate client transactions with a server transaction for forking proxies or B2BUAs. Upon forking, implementations can reuse the value they inserted into the topmost Via header’s branch parameter, or they can generate a unique identification string for the client transaction. A more detailed explanation of why it is needed is provided next.

SIP Proxies may fork, creating several client transactions that correlate to a single server transaction. Responses arriving on these client transactions, or new requests (CANCEL, ACK) sent on the client transaction need log file entries that correlate with a server transaction. Similarly, a B2BUA may create one or more client transactions in response to an incoming request. These transactions will require correlation as well.

To best demonstrate the correlation directives "server-txn" and "client-txn", some examples are necessary. In order to do so, it helps to use a canonical representation for the SIP CLF. The most expedient way to do so is to use an ASCII representation for illustration purposes only, but to be safe, the working group should okay this since a specific SIP CLF format has not been defined yet. To get a gist of how these correlation directives help, please see Section 6 of a predecessor [5] to this draft.

Finally, the SIP CLF should be extensible such that future SIP methods, headers and bodies can be represented as well.
9. Security Considerations

A log file by its nature reveals the both the state of the entity producing it and the nature of the information being logged. To the extent that this state should not be publicly accessible and that the information is to be considered private, appropriate file and directory permissions attached to the log file should be used. In the worst case, public access to the SIP log file provides the same information that an adversary can gain using network sniffing tools (assuming that the SIP traffic is in clear text.) If all SIP traffic on a network segment is encrypted, then special attention must be directed to the file and directory permissions associated with the log file to preserve privacy such that only a privileged user can access the contents of the log file.

Transporting SIP CLF files across the network pose special challenges as well. While transporting SIP CLF files is out of scope in the current working group charter, it seems worth drawing attention to the fact that if the file is transported using unencrypted FTP or email, intermediaries and adversaries may have access to the raw SIP CLF records. Accordingly, if the SIP CLF file is to be moved from the generating host, secure FTP or secure email must be used instead.

The SIP CLF represents the minimum fields that lend themselves to trend analysis and serve as information that may be deemed useful. Other formats can be defined that include more headers (and the body) from Section 8. However, where to draw a judicial line regarding the inclusion of non-mandatory headers can be challenging. Clearly, the more information a SIP server logs, the longer time the logging process will take, the more disk space the log entry will consume, and the more potentially sensitive information could be breached. Therefore, adequate tradeoffs should be taken in account when logging more fields than the ones recommended in Section 8.

Implementers need to pay particular attention to buffer handling when reading or writing log files. SIP CLF entries can be unbounded in length. It would be reasonable for a full dump of a SIP message to be thousands of octets long. This is of particular importance to CLF log parsers, as a SIP CLF log writers may add one or more extension fields to the message to be logged.
10. Operational guidance

SIP CLF log files will take up substantive amount of disk space depending on traffic volume at a processing entity and the amount of information being logged. As such, any enterprise using SIP CLF should establish operational procedures for file rollovers as appropriate to the needs of the organization.

Listing such operational guidelines in this document is out of scope for this work.

11. IANA Considerations

This document does not require any considerations from IANA.

12. Acknowledgments

Members of the sipping, dispatch, ipfix and syslog working groups provided invaluable input to the formulation of the draft. These include Benoit Claise, Spencer Dawkins, David Harrington, Christer Holmberg, Hadriel Kaplan, Atsushi Kobayashi, Jiri Kuthan, Scott Lawrence, Simon Perreault, Adam Roach, Dan Romascanu, Robert Sparks, Brian Trammell, Dale Worley, Theo Zourzouvillys and others that we have undoubtedly, but inadvertently, missed.

Adam Roach proposed and documented the binary TLV-format and Hadriel Kaplan proposed and documented the PCAP-based format as alternative SIP CLF representations.

13. References

13.1. Normative References


13.2. Informative References


Authors' Addresses

Vijay K. Gurbani (editor)
Bell Laboratories, Alcatel-Lucent
1960 Lucent Lane
Naperville, IL 60566
USA

Email: vkg@bell-labs.com

Eric W. Burger (editor)
Neustar, Inc.
USA

Email: eburger@standardstrack.com
URI: http://www.standardstrack.com
Tricha Anjali
Illinois Institute of Technology
316 Siegel Hall
Chicago, IL  60616
USA
Email: tricha@ece.iit.edu

Humberto Abdelnur
INRIA
INRIA - Nancy Grant Est
Campus Scientifique
54506, Vandoeuvre-les-Nancy Cedex
France
Email: Humberto.Abdelnur@loria.fr

Olivier Festor
INRIA
INRIA - Nancy Grant Est
Campus Scientifique
54506, Vandoeuvre-les-Nancy Cedex
France
Email: Olivier.Festor@loria.fr