Session ID Draft Outline

1 ABSTRACT 2

2 INTRODUCTION 2

3 TERMINOLOGY 3

4 BINDING OBJECTIVES (FOR WHAT REASONS DO WE WANT TO BIND THINGS TOGETHER) 4

5 BINDING CONCEPTS 5

6 RELATED WORK 5

6.1 DIAL-IN RADIUS 5

6.2 DIAL-IN DIAMETER 6

6.3 WWW BASED SERVICES (CONTENT, MESSAGING, SHOPS) 6

7 SESSION ID REQUIREMENTS 7

8 GENERATION OF SESSION IDS 8

9 AUDITING 8

10 ACCOUNTING 8

11 ANONYMOUS SERVICE USAGE 9

12 EXAMPLES 9

13 SECURITY CONSIDERATIONS 9

14 REFERENCES 9

Author List:
Georg Carle
John Vollbrecht
Sebastian Zander
Tanja Zseby

1 Abstract

This draft describes bindings that are needed for auditing and accounting in AAA. It introduces the concept of a session ID. The session ID is used to tie together a set of related activities. These activities can be single messages, transactions or sub-sessions.
Accounting Records have to be tied together and linked to a user ID in order to produce a bill. Auditing requires a binding of authentication, authorization and accounting to the actual service of one session in order to trace back all occurred actions. A further useful binding is the coupling of different sub-sessions (e.g. video- and audiosession) to a super-session (e.g. the videoconference).

2 Introduction

The Generic AAA Architecture described in [6] is an infrastructure that supports authentication, authorization and accounting for generic services. This draft describes the concept of a session id which is needed in [6] for:

- Binding of Authentication, Authorization and Accounting with the Service provisioning process (Service Session)
- Gathering together accounting records (maybe generated by different hosts) which provide the accounting data for the services a user has used
- Linkage between different service sessions that belong together for auditing and accounting

A session is linked to one or more "users" which might be an individual, an IP address, a hardwareor software port. A user may act as a customer or with the permission of the customer. The user might be identified with a public key, a user-id, an IP address, a phone number. A session also relates to one or more service "providers" which support the session with resources of some sort. Furthermore a session may incorporate one or more "broker" which supports information on providers to use.

A session should have a session id which allows each provider and user in the session to audit session activity and compare it with what other providers and users believe happened in the session. The session id should allow a provider to merge accounting data for the different activities and generate a bill. Each user and provider may have a different view on the overall session. In particular a user or provider may only know a part of the session that took place.

Authentication, authorization and service usage needs to be linked to make a later auditing and accounting possible. This is even more important if these functions are performed by different entities (in different domains). In the case of fraud it must be checked whether the authentication authenticated a wrong person or the rights granted by the authorization process were wrong. The service usage must be linked to authentication and authorization to associate the accounted data in the service session to the person which requested the service and has been identified during the authentication. Furthermore in the case of misuse it must be possible to backtrack the malicious service user which hopefully could be done having a link to the authentication and authorization.

In case we have more than one accounting record generated per session (service) the session id is used to link the different records together. In [2] it is shown that even in the simple dialup service we haven at least two accounting records (start and stop) which must be tied together to provide useful information for later billing. Other services may well produce more

than two records for example if interim reports are used or we have different service equipment involved (e.g. print service).

A service which is provided with the architecture described in [6] may be composed of different services from different providers. For the user this may still look like a single service. To correlate accounting and auditing information the different sessions which are part of a service need to be linked together. This can be achieved via session ids. Several alternatives for linking are possible. One alternative would be the use of a common session id. Another alternative would be to use individual session ids for each subsession, in combination with a binding function (or a binding service) that would allow to obtain information about the related services. In the accounting case it might not be necessary to link to the session ID if there is a separate billing for each accounted process.

A session id must be unique for each provider-user combination. This means that each provider and user may have to create a piece of the session id to guarantee that it is unique to it. It is also possible that the provider creates the whole session id by using user and provider information.

3 Terminology

Service
A service is a performance offered by a provider. In contrast to goods, services are usually intangible and production and consuming happens simultaneously [britannica.com]
Services can be very different. A service can be for instance a 2 hour videoconference including audio, video and whiteboard, the guaranteed transmission of 100 000 packets with a high priority, the provisioning of accounting records, etc.

Session
A session is a service provisioning over time. Start and stop of the session is determined by the service definition. Examples for sessions are audio/video flows, downloading data, collecting continuous (interim) accounting data, etc. Also the concatenation of transactions can be called a session //?not sure about this...//. A session may have subsessions and transactions. A session has three different phases: session establishment, service session, session teardown.

Super- and Subsessions
A session can be a super- or subsession. A session is called a supersession if it has different subsessions. A subsession is a session which is part of a supersession. Whether a particular session is a super- or subsession depends on the viewpoint (e.g. a supersession of a subsession can be a subsession of another (super)session). The supersession should have knowledge of all subsessions. In certain cases it also might be desirable to be able to determine the supersession ID from the subsession ID. Nevertheless, for privacy reasons knowledge about the subsession should not automatically imply knowledge about the supersession. Privacy may be compromised if the id of the supersession contains some cleartext information which might for instance reveal participants. Even if the supersession encrypts these information even the knowledge that there exists a supersession may be problematic. (e.g. even if the communication between A and B is encrypted the fact that someone knows that A communicates with B may be already compromising for A and/or B).

Transaction
A transaction consists of request and response messages that are exchanged. Transactions differ from sessions because they do not require a “continuous” flow of data. Examples for transactions are the authorization of a service, buying a book, etc. Transactions can be part of a session. Simple transactions consist of a single request-response message pair. An Authentication and/or authorization transaction is often used to determine if a session is allowed prior to actual session establishment.

Authentication Transaction
An authentication transaction is the exchange of request/response messages to perform authentication.

Authorization Transaction
An authorization transaction is the exchange of request/response messages to perform authorization. During a session reauthorization transactions might be necessary.

Accounting Transaction
An accounting transaction is the exchange of request/response messages to perform accounting. Accounting can be performed in the form of accounting transactions that report on resource usage by a session. Accounting transaction can occur during a session if accounting or charging indications are needed [pol based acct] or only at the start and the end of the session.

Accounting Session
An accounting session consists of subsequent accounting records. It can be seen as the concatenation of single messages or as concatenation of multiple accounting transactions.

Auditing
Auditing is the process of examining information on a provided services to check whether the service has been provided correctly and the contractual negotiated parameters have been met. Auditing should be performed by an independent third party in order to prevent fraud.

Auditing Transactions
Auditing transactions are message exchanges used in the process of auditing a session. This could be requests to store or to query auditing information.

Peer Session
A peer-session is a session which is on the same level with another session (this means it is neither a super- nor a sub-session). Peer-sessions may interact which each other.

AAA Transactions
AAA messages and transactions may be considered a form of signaling used to control a session. AAA transactions should have unique transaction ids, different from the session id, which can be used to associate the transactions with the session.

4 Binding Objectives (for what reasons do we want to bind things together)

There are different objectives to bind messages, sessions and transactions together.

A) Binding authentication, authorization transactions to the main service session and accounting: This is required in order to concatenate information about the user (user ID) to the session and the involved transactions. For accounting it is important to map the user ID to the corresponding accounting records. For auditing it is also of interest how the user has been authenticated (e.g. strong or weak mechanism) and authorized.

B) Binding together accounting records for one service session: This is needed to recognize which accounting records belong to the same service session. The bound accounting records form the accounting session.
C) Binding together sub-sessions which belong to a service-package or super-session: This would be for example the binding of audio- and videosession for a videoconference (super-session). Binding of sub-sessions to a super-session is useful to concatenate the services used by one user simultaneously and the accounting data for this session.

// comment: we could think about naming the IDs in according to the bindings they provide. I would propose: A) auditing ID, B) accounting ID and C) session ID or session grouping ID //

5 Binding Concepts

In the following we describe two different concepts for binding messages, sessions and transactions together:

- Hierarchical Binding: With an hierarchical binding sub-session ID are derived from the super-session (or parent-session) ID. Here we have to distinguish whether one should be able to determine the supersession ID from subsession IDs. For instance whether the ID of a videoconference session could be derived from the audio-session ID. This might violate privacy requirements.
- Peer-to-peer Binding: With a peer-to-peer binding two sessions at an equal level are concatenated (e.g. an audio-session that consists of two audio-sessions in different provider networks). Information on the binding between two sessions can be stored at the concatenation points where both sessions are known (e.g. AAA server or the border routers between two providers).

6 Related Work

This chapter gives an overview of existing services which already use the concept of a session id.

6.1 Dial-in RADIUS

[1] defines the RADIUS protocol which is used for authorization and accounting a dial-in service. A RADIUS session is defined as follows: Each service provided by the NAS to a dial-in user constitutes a session, with the beginning of the session defined as the point where service is first provided and the end of the session defined as the point where service is ended. A user may have multiple sessions in parallel or series if the NAS supports that.

For accounting purposes a session id is generated [2]. RADIUS generates on accounting records at the start and the stop of a session. The RADIUS Acct-Session-ID attribute is used to match the two reports of a session and is defined as follows: This attribute is a unique Accounting ID to make it easy to match start and stop records in a log file. The start and stop records for a given session MUST have the same Acct-Session-ID. It is strongly recommended that the Acct-Session-ID be a printable ASCII string. For example, one implementation uses a string with an 8-digit upper case hexadecimal number, the first two digits increment on each reboot (wrapping every 256 reboots) and the next 6 digits counting from 0 for the first person logging in after a reboot up to 2^24-1, about 16 million. Other encodings are possible.

Another attribute which is defined in [2] is the Acct-Multi-Session-ID. This attribute links together multiple related sessions and thus can be seen as kind of supersession id with regard to the session id of one session. The attribute is defined as follows: This attribute is a unique Accounting ID to make it easy to link together multiple related sessions in a log file. Each session linked together would have a unique Acct-Session-ID but the same Acct-Multi-Session-ID. It is strongly recommended that the Acct-Multi-Session-ID be a printable ASCII string.

6.2 Dial-in DIAMETER

The successor of RADIUS is the basis for the forthcoming AAA protocol DIAMETER [3]. DIAMETER has a session id for binding the different AAA transaction together. Therefore a Session-ID AVP has been defined. The initial request for authentication and/or authorization of a user would include the Session-ID. The Session-ID is then used in all subsequent messages to identify the user's session. The session state (associated with a Session-ID) is freed upon receipt of the Session-Termination-Request, Session-Termination-Answer and according to rules established in a particular extension/application of DIAMETER.

The Session-ID AVP is defined as follows: The Session-ID AVP (AVP Code 263) is of type Data and is used to identify a specific session (see section 3.0). All messages pertaining to a specific session MUST include only one Session-ID AVP and the same value MUST be used throughout the life of a session. When present, the Session-ID SHOULD appear immediately following the DIAMETER Header. For messages that do not pertain to a specific session, multiple Session-ID AVPs MAY be present as long as they are encapsulated within different Grouped-AVP AVPs [messages which do not belong to any session!]. The Session-ID MUST be globally unique at any given time since it is used by the server to identify the session (or flow). The format of the session identifier SHOULD be as follows:

<Sender's Host-Name><sender's port number> <monotonically increasing 32 bit value><optional value>

The monotonically increasing 32 bit value SHOULD NOT start at zero upon reboot, but rather start at a random value. This will minimize the possibility of overlapping Session-IDs after a reboot. Alternatively, an implementation MAY keep track of the increasing value in non-volatile memory. The optional value is implementation specific but may include a modem's device ID, a layer 2 address, timestamp, etc. The session ID is created by the DIAMETER device initiating the session, which in most cases is done by the client. Note that a Session-ID MAY be used by more than one extension (e.g. authentication for a specific service and accounting, both of which have separate extensions).

The Session-Timeout AVP can be used by a server to tell the client the maximum session duration. The client may use this attribute to indicate the maximum length that it is willing to accept.

6.3 WWW based services (content, messaging, shops)

HTTP is a stateless protocol. However for the provision of certain services there is need to keep state over a number of different HTTP requests. A session id is used in web based services to allow the following: tracking of user, keep state for a user over different pages/http requests, limited form of authentication (prevent replay attacks). There are two traditional methods of propagating a session id which have been used before HTTP state management was invented [5]: URL parameter or cookies. Cookies are optimal but not reliable since the client can disable the use of them. A session id used for authentication is normaly a short-lived cookie which is only kept in browser memory and is not written to disk. However if

used for keeping state the session id may be stored on disk, so that the state can be resumed even after shutoff and restart of the computer.

[5] describes a way to create stateful sessions with HTTP requests and responses. The state management mechanism allows clients and servers that wish to exchange state information to place HTTP requests and responses within a larger context, which is called a "session". This context might be used to create, for example, a "shopping cart", in which user selections can be aggregated before purchase, or a magazine browsing system, in which a user's previous reading affects which offerings are presented. Neither clients nor servers are required to support cookies. A server MAY refuse to provide content to a client that does not return the cookies it sends.

In [4] it is shown that the HTTP State Management mechanism is both useful and controversial compared to the traditional methods. It is useful because numerous applications of HTTP benefit from the ability to save state between HTTP transactions, without encoding such state in URLs. It is controversial because the mechanism has been used to accomplish things for which it was not designed and is not well-suited. Some of these uses have attracted a great deal of public criticism because they threaten to violate the privacy of web users, specifically by leaking potentially sensitive information to third parties such as the Web sites a user has visited. There are also other uses of HTTP State Management which are inappropriate even though they do not threaten user privacy.

7 Session ID Requirements

Global uniqueness:
To uniquely identify a session a session id must be globally unique which means unique in time and location. The might be relaxed to unique within a given time period or within a certain spatial scope. For instance if we have to audit session data for 10 years the session id must be unique within this period. If it is assured that a session does not cross certain boundaries the uniqueness requirement may be relaxed to “unique within this boundaries”.

Privacy:
A session id must not enable a third party to be able to derive the "real" id of the user from the session id. The session id should only be resolvable to the entity which generated it. For the provisioning of anonymous services it might be required that the session ID is nor resolvable at all. In case of a fully anonymous session the session id has to be build without even knowing user specific parameter. Therefore no relation to the user is given. Furthermore no information should be logged which would give a hint on the real id.

Efficiency:
The generation and linkage of session ids must be sufficiently fast compared to the overall process of service authentication and authorization to avoid unnecessary latency. Especially it must be avoided to increase latency in service provision due to the transport of session ids between providers.

Security:
A session id must be non predictable. Otherwise it may be possible to take over another persons session. It must not be possible to derive user data from the session id. This can be avoided by using a one way hash over the data (e.g. [MD5]). There may be situations where it is allowed to directly use user data in the session id. For security reasons session IDs could be transmitted via encrypted channels (e.g. IPSEC).

Flexibility:
There may be different solutions of generating/assigning session id and binding session ids together which can be used in different situations. One method can be tailored to a specific service (usage). However all methods must be compatible in a way such that accounting and auditing is not limited to an unacceptable degree.

Granularity:
The solution must allow for the finest grained auditing possible of a provided service. However cases in which there is only lax auditing required should also be supported efficiently.

8 Generation of Session IDs

When and where should the session ID be generated?
To tie the three As together the session ID must be generated as part of the authentication process. The session ID may be build using parameter from the authentication request as well as IP address or DNS name of the requester. The authentication may be performed by the service equipment (e.g. dialup the NAS) or by another entity. In case of a free service there may be no authentication and authorization. In this case the session ID is generated by the service equipment at the start of the service usage.

How is the Session ID generated?
A global unique session id can created in the generic AAA architecture as follows. The id is composed of three different parts: id = user_id+service_id+AAA_id
The user_id is an id which is created by the service. This id may be based on user specific information. The id may contain user specific information in cleartext or it may be a cryptographic hash over some user information. The service_id is the id which is used to identify the service at the AAA server. The AAA server needs some kind of id to map a certain request to a certain service i.e. pass the request to the ASM. The AAA_id is a global unique id of the AAA server providing the requested service. The id of the AAA needs to be globally unique anyway.

9 Auditing

Auditing is used to examine the correct provisioning of the service. The auditing process should be defined in a way that it is accepted by both sides (provider and user) as a valid process to proof that the provider has fulfilled the provisioning task correctly even in a legal action.
A session ID is required here to map all actions, messages, transactions and sub-sessions to the specific user. Since Auditing requires traceability of all performed actions, the auditability of a service is usually in direct contrast to anonymity. For privacy reasons it is possible to restrict access to the data used for auditing to a few trusted instances and/or to distribute the knowledge of the concatenated actions in a way that only the concatenation of information from different instances gives a the whole view.

A session id is always needed except we provide no auditing function at all. This may be the case for a cheap service. Auditing capability may be an added value for the customer. On the other hand the government may force providers to audit specific activities and/or data.

10 Accounting

The binding of accounting records to a specific user is required to generate a bill. Furthermore a session ID is used to tie together accounting messages and transactions. This can be subsequent records for one session, records form different sources or providers.

11 Anonymous Service Usage

The ability to provide an anonymous services is very important. The anonymity of the service user can prevent proper auditing and therefore is in direct contrast to the requirement for auditability of the service. We distinguish between a full anonymous services where it is not possible at all to obtain the user-session mapping and a partial anonymous services where certain instances are able to resolve the user-session mapping.

12 Examples

Show how this works

Should cover:
Accounting, auditing, session start, session teardown, adding/deleting new subsession

Video on Demand (with QoS(Diffserv))
VPN (with QoS)
Content service (with QoS)
Dialup (with Mobility/Roaming)
Bandwidth request
Network printing
E-commerce
Anonymous service

13 Security Considerations

Privacy
Forging Session IDs (spoofing for hacking a session)
Encypted Session IDs

14 References

[1] Rigney C., Rubens A., Simpson W. and S. Willens, "Remote Authentication Dial In User Service (RADIUS)", RFC 2138, April 1997.

[2] Rigney, C., "RADIUS Accounting", RFC 2139, April 1997.

[3] P. Calhoun, A. Rubens, H. Akhtar, E. Guttman. "DIAMETER Base Protocol", draft-calhoun-diameter-17.txt, IETF work in progress, September 2000.

[4] K. Moore, N. Freed: " Use of HTTP State Management", IETF RFC2964, October 2000

[5] D. Kristol, L. MontulliRFC: "HTTP State Management Mechanism", IETF RFC2965, October 2000

[6] C. de Laat, G. Gross, L. Gommans, J. Vollbrecht, D. Spence, "Generic AAA Architecture", draft-irtf-aaaarch-generic-01.txt, Work in Progress, March 2000

[MD5] R. Rivest: "The MD5 Message-Digest Algorithm" IETF RFC1321, April 1992