Network Working Group
Internet-Draft
Intended status: Informational
Expires: March 2, 2007

M. Wahl
Informed Control Inc.

August 29, 2006

# LDAP Session Tracking Control
# draft-wahl-ldap-session-01

## Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of 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 March 2, 2007.

## Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

Many network devices, application servers, and middleware components of a enterprise software infrastructure generate some form of session tracking identifiers, which are useful when analyzing activity and accounting logs to group activity relating to a particular session. This document discusses how Lightweight Directory Access Protocol version 3 (LDAP) clients can include session tracking identifiers with their LDAP requests. This information is provided through controls in the requests the clients send to LDAP servers. The LDAP server receiving these controls can include the session tracking identifiers the the log messages it writes, enabling LDAP requests in the LDAP server's logs to be correlated with activity in logs of other components in the infrastructure. The control also enables session tracking information to be generated by LDAP servers and returned to clients and other servers. Three formats of session tracking identifiers are defined in this document.

## 1. Introduction

The majority of directory server implementations produce access logs detailing each request they receive. These logs can be read using log parsing tools or specialized log viewer applications. Typically it will be possible, for each request logged by a directory server, to determine the bind DN (or possibly another form of authentication identity) of the client which sent the request to the server, and many servers also log the IP address of the client that sent the request.

In the original OSI architecture, it was envisaged that users might interact with a directory service through specialized applications, known as Directory User Agents, that were the clients of the Directory Access Protocol. Similarly, in early Internet directory deployments, a majority of LDAP clients were desktop applications, that used the LDAP protocol to search an enterprise directory for address book/contact information.

Today, the majority of LDAP clients are embedded within middleware and server applications. Legacy address book protocols might be gatewayed into LDAP, or a server might consult an LDAP server in order to check a user's password or obtain their preferences. While the LDAP requests might result from a user's activity somewhere on the network, it is rare for the user to be 'driving' the LDAP client, and in most cases the user performing the activity is unaware that LDAP requests are being generated on their behalf.

However, this information is important to directory system administrators and auditors. They may wish to determine who is making use of the directory service, or track the source of unusual requests.

When a directory server administrator reviews a log file produced by a directory server that has been accessed only by clients that are themselves middleware, where the end user does not interact with the middleware directly, only through other kinds of servers (e.g. application servers or remote access servers), it will be difficult to correlate between the directory server's log and the logs of the servers which made use of this directory to determine why the LDAP requests were made and who were responsible for causing them.

Reasons for this include:

- Directory servers are capable of performing many hundreds of requests per second or more, and even with time synchronization between the systems on which the directory server and middleware are deployed, times of requests might not be logged accurately enough to be able to correlate based on time: the server's logs might be only to 1-second resolution.

- a single function on a middleware server, such as "authenticate a user", may result in multiple LDAP requests being generated in order to perform that request.

- Many high performance middleware servers implement connection pooling, managing a set of persistent connections to each directory server and multiplexing operations across the connections. Each connection will have the same source IP address and bind DN. If a particular activity causes multiple LDAP requests to be generated, each LDAP request might be sent on a different connection. Also, as LDAP is an asynchronous protocol, middleware servers may have more than one request in progress on each connection, asynchronously sending requests to the directory server on each connection and processing the responses in whatever order they are received.

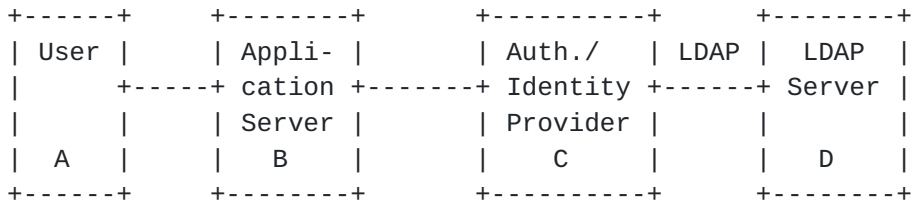
This document defines a new control for use in LDAPv3 [1] operation requests. This control contains session tracking information that can be used to correlate log information present in the directory server's log with the logs of other middleware servers.

The words "MUST", "SHOULD" and "MAY" are used as defined in RFC 2119 [2].

Please send comments to the author at mark.wahl@informed-control.com.

## 2. Session tracking

A typical enterprise deployment with a an application indirectly relying upon the directory might resemble:




In this diagram, a user (A) makes some request of an application server (B). The application server might rely on an integrated or external authentication provider in order to check the user's authentication credentials, or might use an identity provider to obtain profile information about the user. This request might be made through an API or a protocol other than LDAP, e.g. RADIUS, Kerberos, SMB, etc. The authentication/identity provider (C) would generate one or more LDAP requests and send them to an LDAP server (D).

The LDAP server has the following information already available to it through the LDAP protocol: the IP address and authentication credentials of (C). If the provider has included the Proxy Authorization Control [11], then it may also receive the Distinguished Name (DN) or authorization identity of either (A) or (B), depending on how (C) uses the directory. In order to obtain this DN, however, (C) might need to perform one or more LDAP search or bind requests. If there is no entry in the directory corresponding to the identity of (A) or (B), then there is no way in the base LDAP specification or the Proxy Authorization Control for (C) to describe (A) or (B) to (D).

If either (B) or (C) have generated a session identifier for tracking the interactions of (A) for a particular session, then it is useful to include this information with the requests made to the directory server, so that this session identifier will show up in the directory server's logs. That is the purpose of the control defined in the next section.

## 3. Description of the control

There is currently no standard way of describing a session: there are many different formats for a session identifier, and each application that tracks sessions typically has its own semantics for what a session means. Thus, a control is defined using an extensible model, in order to incorporate many different application's concepts and formats of a session tracking identifier.

The value of the session identifier control encapsulates the following four pieces of information: sessionSourceIp, sessionSourceName, formatOID and sessionTrackingIdentifier.

The sessionSourceIp field is a US-ASCII string encoding of an IPv4 or IPv6 [3] address of the component of the system which has generated a session tracking identifier. The purpose of this field is to enable the directory server administrator, even if they do not have a log parser that understands a particular session tracking identifier format, to at least be able to identify the server that manages the session. Note that there is no guarantee of IP-level connectivity between the directory server and the system which generated the tracking identifier, and if Network Address Translation is being used, the IP address in this field might be from a private use address range.

The sessionSourceName field is a UTF-8 [4] encoded ISO 10646 [5] text string. This field describes the component of the system which has generated a session tracking identifier. The format of this field is determined by the formatOID (discussed below); examples of contents of a sessionSourceName field might be a hostname, a distinguished name, or a web service address. This contents of this field is not intended to identify an end user; instead it identifies the server using a naming scheme other than IP address.

The formatOID is a US-ASCII encoded dotted decimal representation of an OBJECT IDENTIFIER. The OBJECT IDENTIFIER indicates the scheme that is used to generate the sessionSourceName and sessionTrackingIdentifier fields. As there is currently no standard scheme for session information, it is expected that there will be many different formats carried within this control. Three possible format OIDs are presented later in this document.

The sessionTrackingIdentifier field is a UTF-8 encoded ISO 10646 string. The session identifier SHOULD be limited to whitespace and printable characters; non-printing and control characters SHOULD NOT be used, and byte sequences that are not legal UTF-8 MUST NOT be used. The syntax of the session identifier and its semantics (e.g., how values are compared for equality) are governed by the formatOID.

For example, the session identifier might be a simple string encoding of a decimal counter, a username, a timestamp, a fragment of XML, or it might be something else, depending on the format.

## 4. Formal definition

The controlType is 1.3.6.1.4.1.21008.108.63.1, the criticality MUST be either FALSE or absent, and the controlValue MUST be present. The controlValue OCTET STRING is always present and contains the bytes of the BER [6] encoding of a value of the ASN.1 data type SessionIdentifierControlValue, defined as follows:

```
LDAP-Session-Identifier-Control
DEFINITIONS IMPLICIT TAGS ::=

BEGIN

SessionIdentifierControlValue ::= SEQUENCE {
     sessionSourceIp                LDAPString,
     sessionSourceName              LDAPString,
     formatOID                      LDAPOID,
     sessionTrackingIdentifier      LDAPString
}

END
```

The sessionSourceIp element SHOULD NOT be longer than 42 characters (necessary for a string representation of an IPv6 address), and MUST NOT be longer than 128 characters. Each character will be encoded into a single byte. If the IP address of the system which generated the session tracking identifier is not known, the sessionSourceIp element SHOULD be of zero length.

The sessionSourceName element SHOULD NOT be longer than 1024 characters, and MUST NOT be longer than 65536 characters. Note that in the UTF-8 encoding a character MAY be encoded into more than one byte. If no other addressing information about that system is known or relevant to the format, the sessionSourceName element SHOULD be of zero length.

The formatOID element MUST contain only the US-ASCII encodings of the ISO 10646 characters FULL STOP and DIGIT ZERO through DIGIT NINE (0x2E, 0x30-0x39). The formatOID element MUST NOT be of zero length, and SHOULD NOT be longer than 1024 characters.

The sessionTrackingIdentifier field MAY be of zero length (although this might not be useful). There is no upper bound on the sessionTrackingIdentifier, but it is suggested that values SHOULD NOT be longer than 65536 characters without prior agreement with the directory server administrator. Note that in the UTF-8 encoding a character MAY be encoded into more than one byte.

## 5. Use in LDAP

The control MAY be included in any LDAP operation. The control has order-dependent semantics.

A client might place the control on a message with a bindRequest, searchRequest, modifyRequest, addRequest, delRequest, modDNRequest, compareRequest or extendedReq. A client MAY include multiple controls of this type in a single request. This enables the client to incorporate multiple distinct session tracking identifiers with different formats.

When a network service is proxying or chaining LDAP, in which the service receives an incoming LDAP request from a client and from this generates one or more requests to other LDAP servers, the service SHOULD include any controls of this type that it received from clients in requests it generates, without modification. A service MAY silently remove controls if the control would violate security policy. If the service has its own session state identifier, it SHOULD include the session identifier control it generates in the Controls SEQUENCE after any session identifier controls received by clients. (If there are multiple proxies involved, each will add their own session state to the end of the controls list).

A server might place the control on message with a bindResponse, searchResDone, modifyResponse, addResponse, delResponse, modDNResponse, compareResponse, extendedResp or intermediateResponse. The server can include the control in the response regardless of whether the client included a control in the request or not. (The control in a response is unsolicited, and a client which does not recognize the control or a session tracking format can safely ignore the control, as discussed in the following section). A server MAY include multiple controls of this type in a response.

## 6. Extensibility considerations

The following sections of this document after this section define 3 possible formats, and it is expected that applications MAY define their own formats to represent session tracking identifiers already implemented.

An application developer or server developer who wishes to transfer their implementation's format for session tracking identifier within an LDAP control MUST choose a new, unique, OBJECT IDENTIFIER to represent this format.

The format determines the semantics of the sessionSourceName string, and the sessionTrackingIdentifier string.

In general, when an LDAP server that has session tracking logging enabled receives one or more of these controls with a request, the server SHOULD include all fields of all of the controls with the logging information for the request.

A LDAP server that supports third-party or extensible log parsing tools SHOULD NOT reject or ignore a control if the formatOID value is not recognized, as it is expected that applications may include session tracking identifiers and want to make this information available to log parsers for correlation purposes, even if the directory server does not need to make any use of this information.

However, if the LDAP server does not recognize the control or if the control is not properly formatted, the LDAP server SHOULD ignore the control and process the request as if the control had not been included.

When an LDAP client receives a response that includes this control, the behavior depends on the client implementation. Clients SHOULD silently ignore controls with formats they do not recognize, and process the response as if the control had not been included.

## 7. Formats for use with RADIUS accounting

This section defines two possible session tracking formats, that can be used in LDAP clients that are part of or used by RADIUS servers [7].

With formatOID set to 1.3.6.1.4.1.21008.108.63.1.1 within the control value, the sessionTrackingIdentifier SHOULD contain the value of the Acct-Session-Id RADIUS attribute (type 44), as defined in RFC 2866 [8]. (RFC 2866 section 5.5 states that the Acct-Session-Id SHOULD contain UTF-8 encoded 10646 characters.)

With formatOID set to 1.3.6.1.4.1.21008.108.63.1.2 within the control value, the sessionTrackingIdentifier SHOULD contain the value of the Acct-Multi-Session-Id RADIUS attribute (type 50), as defined in RFC 2866 [8]. (RFC 2866 section 5.11 states that the Acct-Multi-Session-Id SHOULD contain UTF-8 encoded 10646 characters.)

In both of these two formats, the value of the sessionSourceIp field SHOULD contain either a string encoding value of the IPv4 address from the NAS-IP-Address RADIUS attribute (type 4), or a string encoding of the IPv6 address from the value of the NAS-IPv6-Address RADIUS attribute (type 95) as defined in RFC 3162 [9]. The value of the sessionSourceName field SHOULD contain a string encoding the value of the NAS-Identifier RADIUS attribute (type 32), if present, or be of zero length if the NAS-Identifier RADIUS attribute was not provided or was not in a recognized format.

## 8. Format for username accounting

This section defines another possible session tracking format, that can be used in LDAP clients that are part of applications which identify users with simple string usernames.

With formatOID set to 1.3.6.1.4.1.21008.108.63.1.3 within the control value, the sessionTrackingIdentifier SHOULD contain a username that has already been authenticated by the application that is generating the session. This format SHOULD NOT be used for purported names, where the application has not verified that the username is valid.

The sessionSourceName field SHOULD contain the hostname where that application is running, or be of zero length if the hostname is not known.

The username SHOULD be a SASL authorization identity string, as described in section 3.4.1 of RFC 4422 [10]. It is expected that these usernames are not globally unique, but are only unique within the context of a particular application or particular enterprise.

A control with this format differs from the Proxied Authorization Control as defined in RFC 4370 [11], as the presence of this session identifier control on a request SHOULD NOT influence the directory server's access control decision of whether or how to perform that request.

Note that this format does not provide any information to differentiate between multiple sessions or periods of interaction by the same user. It is primarily intended for deployments which merely need to be able to tie each directory operation to they identity of the user whose activities caused the operation request to be generated, even if the user might not even be represented in the directory where the operations are being performed.

For example, if an application server "app.example.com" with IPv4 address "192.0.2.1" had authenticated an user with name "bloggs", and then sent a search request to the LDAP directory in order to obtain some public information on service configuration intending to provide it to that user, the application might include a session identifier control, with controlType 1.3.6.1.4.1.21008.108.63.1, criticality FALSE, and the controlValue the BER encoding of the following ASN.1 value:

```
      {  -- SEQUENCE
         "192.0.2.1",                    -- sessionSourceIp
         "app.example.com",              -- sessionSourceName
         "1.3.6.1.4.1.21008.108.63.1.3", -- formatOID
         "bloggs"                        -- sessionTrackingIdentifier
      }
```

## 9. Security Considerations

The session identifier controls used in this document are not intended as a security control or proxy authentication mechanism, and SHOULD NOT be used within the server to influence behavior.

Malicious clients might attempt to provide false or misleading information in directory server logs through the use of this control. LDAP servers SHOULD implement access checks which limit whether session identifier information provided by a client is logged. These checks might include validating that the request is received from an authenticated client, or perhaps that the client is authorized to use related controls, such as the Proxied Authorization Control [11]. Session identifier information from clients which do not meet the server's access check requirement SHOULD be silently discarded.

In some formats, session tracking identifiers may contain personal-identifiable information, such as usernames or client IP addresses. Unless data link, network or transport level encryption is being used, this information might be visible to attackers monitoring the network segments across which this information is being transmitted.

Correlation of activities across multiple servers can enable administrators and monitoring tools to construct a more accurate picture of user behavior. In particular, this tracking control could be used to determine the set of applications and services with which a particular user has had interactions. Thus, this control would not be appropriate to deployments intending to anonymize directory requests. Session formats containing personal identifiable information SHOULD NOT be used between systems in different organizations where there is no existing agreement between those organizations on privacy protection.

## 10. IANA Considerations

This control will be registered as follows:

Subject: Request for LDAP Protocol Mechanism Registration

Object Identifier: 1.3.6.1.4.1.21008.108.63.1

Description: Session Tracking Identifier

Person & email address to contact for further information:
Mark Wahl <Mark.Wahl@informed-control.com>

Usage: Control

Specification: (I-D) RFC XXXX

Author/Change Controller: Mark Wahl

The OBJECT IDENTIFIER for particular session identifier formats defined for other applications need not be registered with IANA.

## 11. Acknowledgments

This control was inspired by conversations with Greg Lavender. Neil Wilson provided useful feedback on this document.

## 12. References

### 12.1. Normative References

[1] Zeilenga, K., "Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map", RFC 4510, June 2006.

[2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, BCP 14, March 1997.

[3] Hinden, R., "IP Version 6 Addressing Architecture", RFC 1884, January 1996.

[4] Yergeau, F., "UTF-8, a transformation format of ISO 10646", RFC 3629, November 2003.

[5] "Universal Multiple-Octet Coded Character Set (UCS) - Architecture and Basic Multilingual Plane, ISO/IEC 10646-1: 1993".

[6] "ITU-T Rec. X.690 (07/2002) | ISO/IEC 8825-1:2002, "Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", 2002.".

[7] Rigney, C., "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

[8] Rigney, C., "RADIUS Accounting", RFC 2866, June 2000.

[9] Aboba, B., "RADIUS and IPv6", RFC 3162, August 2001.

[10] Melnikov, A., "Simple Authentication and Security Layer (SASL)", RFC 4422, June 2006.

### 12.2. Informative References

[11] Weltman, R., "Lightweight Directory Access Protocol (LDAP) Proxied Authorization Control", RFC 4370, February 2006.

## Appendix A. Copyright

Copyright (C) The Internet Society 2006. This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights. This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Author's Address

Mark Wahl
Informed Control Inc.
PO Box 90626
Austin, TX 78709
US

Email: mark.wahl@informed-control.com

## Full Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

## Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).