INTERNET-DRAFT
draft-ietf-ldapbis-authmeth-13.txt
Obsoletes: 2829, 2830
Intended Category: Draft Standard

Editor: R. Harrison
Novell, Inc.

October, 2004

LDAP: Authentication Methods and Connection Level Security Mechanisms

Status of this Memo

http://www.ietf.org/shadow.html.

Copyright Notice

Abstract

This document describes authentication methods and connection level security mechanisms of the Lightweight Directory Access Protocol (LDAP).

This document details establishment of TLS (Transport Layer Security) using the StartTLS operation.

This document details the simple Bind authentication method including anonymous, unauthenticated, and plain-text password mechanisms and the SASL (Simple Authentication and Security Layer) Bind authentication method including DIGEST-MD5 and EXTERNAL mechanisms.

This document discusses various authentication and authorization states through which a connection to an LDAP server may pass and the actions that trigger these state changes.

Table of Contents

1. Introduction

The Lightweight Directory Access Protocol (LDAP) [Roadmap] is a powerful protocol for accessing directories. It offers means of

searching, retrieving and manipulating directory content, and ways to access a rich set of security functions.

It is vital that these security functions be interoperable among all LDAP clients and servers on the Internet; therefore there has to be a minimum subset of security functions that is common to all implementations that claim LDAP conformance.

Basic threats to an LDAP directory service include:

(1) Unauthorized access to directory data via data-retrieval operations,

(2) Unauthorized access to directory data by monitoring others' access,

(3) Unauthorized access to reusable client authentication information by monitoring others' access,

(4) Unauthorized modification of directory data,

(5) Unauthorized modification of configuration information,

(6) Denial of Service: Use of resources (commonly in excess) in a manner intended to deny service to others,

(7) Spoofing: Tricking a user or client into believing that information came from the directory when in fact it did not, either by modifying data in transit or misdirecting the client's connection. Tricking a user or client into sending privileged information to a hostile entity that appears to be the directory server but is not. Tricking a directory server into believing that information came from a particular client when in fact it came from a hostile entity, and

(8) Hijacking: An attacker seizes control of an established protocol session.

Threats (1), (4), (5), (6), (7) are (8) are active attacks. Threats

(2) and (3) are passive attacks.

Threats (1), (4), (5) and (6) are due to hostile clients. Threats (2), (3), (7) and (8) are due to hostile agents on the path between client and server or hostile agents posing as a server, e.g. IP spoofing.

LDAP offers the following security mechanisms:

(1) Authentication by means of the Bind operation. The Bind operation provides a simple method which supports anonymous, unauthenticated, and authenticated with password mechanisms, and the Secure Authentication and Security Layer (SASL) method which supports a wide variety of authentication mechanisms,

(2) Mechanisms to support vendor-specific access control facilities (LDAP does not offer a standard access control facility)

(3) Data integrity protection by means of security layers in TLS or SASL mechanisms,

(4) Data confidentiality protection by means of security layers in TLS or SASL mechanisms,

(5) Server resource usage limitation by means of administrative limits configured on the server, and

(6) Server authentication by means of the TLS protocol or SASL mechanism.

LDAP may also be protected by means outside the LDAP protocol, e.g. with IP-level security [RFC2401].

At the moment, imposition of access controls is done by means outside the scope of LDAP.

Considering the above requirements, experience has shown that simply allowing implementations to pick and choose among the possible alternatives is not a strategy that leads to interoperability. In the absence of mandates, clients will continue to be written that do not support any security function supported by the server, or worse, they will support only clear text passwords that provide inadequate security for most circumstances.

It is desirable to allow clients to authenticate using a variety of mechanisms including mechanisms where identities are represented as distinguished names [X.501] [Models] in string form [LDAPDN] or are used in different systems (e.g. user name in string form). Because some authentication mechanisms transmit credentials in plain text form and/or do not provide data security services, it is necessary to ensure secure interoperability by identifying a mandatory-to-implement mechanism for establishing transport-layer security services.

The set of security mechanisms provided in LDAP and described in

this document is intended to meet the security needs for a wide range of deployment scenarios and still provide a high degree of interoperability among various LDAP implementations and deployments. Appendix B contains example deployment scenarios that list the mechanisms that might be used to achieve a reasonable level of security in various circumstances.

1.1. Relationship to Other Documents

This document is an integral part of the LDAP Technical Specification [Roadmap].

This document obsoletes RFC 2829.

Sections 2 and 4 of RFC 2830 are obsoleted by [Protocol]. The remainder of RFC 2830 is obsoleted by this document.

1.2. Conventions Used in this Document

1.2.1. Glossary of Terms

The following terms are used in this document. To aid the reader, these terms are defined here.

- "user" represents any human or application entity which is accessing the directory using a directory client. A directory client (or client) is also known as a directory user agent (DUA).

- "connection" refers to the underlying transport protocol connection used to carry the protocol exchange.

- "TLS connection" refers to an LDAP connection with TLS protection [TLS].

- "association" refers to the association that exists between the connection to its current authorization state. As a shorthand, an association with an authorization state of <state> can be referred to as a "<state> association", e.g. an association with an anonymous authorization state is an anonymous association.

1.2.2. Security Terms and Concepts

In general, security terms in this document are used consistently with the definitions provided in [RFC2828]. In addition, several terms and concepts relating to security, authentication, and authorization are presented in Appendix C of this document. While the formal definition of these terms and concepts is outside the scope of this document, an understanding of them is prerequisite to understanding much of the material in this document. Readers who are unfamiliar with security-related concepts are encouraged to review Appendix C before reading the remainder of this document.

1.2.3. Keywords

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 [RFC2119].

2. Implementation Requirements

LDAP server implementations MUST support the anonymous authentication mechanism of simple bind (as discussed in Section 6).

LDAP implementations that support any authentication mechanism other than the anonymous authentication mechanism of simple bind MUST support the DIGEST-MD5 [DIGEST-MD5] mechanism of SASL bind (as detailed in section 11). DIGEST-MD5 is a reasonably strong

authentication mechanism that provides (mandatory-to-implement) data security (data integrity and data confidentiality) services.

LDAP impementations SHOULD support the simple (DN and password) authentication mechanism of simple bind (as detailed in section 8). Implementations that support this mechanism MUST be capable of protecting it by establishment of TLS (as discussed in section 3) or other suitable suitable data confidentiality and data integrity protection (e.g. IPSec).

Implementations MAY support additional authentication mechanisms. Some of these mechanisms are discussed below.

LDAP server implementations SHOULD support client assertion of authorization identity via the SASL EXTERNAL mechanism (sections 3.2.2 and 9).

LDAP server implementations SHOULD support the StartTLS operation, and server implementations that do support the StartTLS operation MUST support the TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA ciphersuite.

3. StartTLS Operation

The Start Transport Layer Security (StartTLS) operation defined in section 4.14 of [Protocol] provides the ability to establish TLS [TLS] on an LDAP connection.

The goals of using the TLS [TLS] protocol with LDAP are to ensure data confidentiality and integrity, and to optionally provide for authentication. TLS expressly provides these capabilities, although the authentication services of TLS are available to LDAP only in combination with the SASL EXTERNAL authentication method (see section 10), and then only if the SASL EXTERNAL implementation chooses to make use of the TLS credentials.

3.1. Sequencing of the StartTLS Operation

This section describes the overall procedures clients and servers must follow for TLS establishment. These procedures take into consideration various aspects of the association including discovery

of resultant security level and assertion of the client's authorization identity.

3.1.1. StartTLS Request

A client may send the StartTLS extended request at any time after establishing an LDAP connection, except:

- when TLS is currently established on the connection,
- when a multi-stage SASL negotiation is in progress on the connection, or
- when it has not yet received responses for all operation requests previously issued on the connection.

As described in [Protocol] Section 4.14.2.2, a (detected) violation of any of these requirements results in a return of the operationsError resultCode.

Client implementers should ensure that they strictly follow these operation sequencing requirements to prevent interoperability issues. Operational experience has shown that violating these requirements causes interoperability issues because there are race conditions that prevent servers from detecting some violations of these requirements due to server hardware speed, network latencies, etc.

There is no general requirement that the client have or have not already performed a Bind operation (section 4) before sending a StartTLS operation request.

If the client did not establish a TLS connection before sending a request and the server requires the client to establish a TLS connection before performing that request, the server MUST reject that request by sending a resultCode of confidentialityRequired.

3.1.2. StartTLS Response

The server will return an extended response with the resultCode of success if it is willing and able to negotiate TLS.

It will return a resultCode other than success (documented in [Protocol] section 4.13.2.2) if it is unwilling or unable to do so. The state of the association is unaffected if a non-success resultCode is returned.

In the successful case, the client (which has ceased to transfer LDAP requests on the connection) MUST either begin a TLS negotiation or close the connection. The client will send PDUs in the TLS Record Protocol directly over the underlying transport connection to the server to initiate [TLS] negotiation.

3.1.3. TLS Version Negotiation

Negotiating the version of TLS to be used is a part of the TLS

Handshake Protocol [TLS]. Please refer to that document for details.

3.1.4. Client Certificate

If an LDAP server requests a client to provide its certificate during TLS negotiation and the client does not present a suitable certificate (e.g. one that can be validated), the server may use a local security policy to determine whether to successfully complete TLS negotiation.

If the client provides a certificate that can be validated, information in the certificate may be used by the server in establishing the client's authorization identity by use of the SASL EXTERNAL mechanism as discussed in Section 9.

3.1.5. Discovery of Resultant Security Level

After a TLS connection is established on an LDAP connection, both parties are to individually decide whether or not to continue based on the security level achieved. The procedure for ascertaining the TLS connection's security level is implementation dependent.

If the client or server decides that the security level is not high enough for it to continue, it SHOULD gracefully close the TLS connection immediately after the TLS negotiation has completed (see [Protocol] section 4.13.3.1 and section 3.2.3 below). The client may then close the connection, attempt to StartTLS again, send an unbind request, or send any other LDAP request.

3.1.6. Server Identity Check

The client MUST check its understanding of the server's hostname against the server's identity as presented in the server's Certificate message in order to prevent man-in-the-middle attacks.

Matching is performed according to these rules:

- The client MUST use the server name provided by the user (or other trusted entity) as the value to compare against the server name as expressed in the server's certificate. A hostname derived from user input is to be considered provided by the user only if derived in a secure fashion (e.g., DNSSEC).

- If a subjectAltName extension of type dNSName is present in the certificate, it SHOULD be used as the source of the server's identity.

- The string values to be compared MUST be prepared according to the rules described in [Matching].

- The "*" wildcard character is allowed. If present, it applies only to the left-most name component.

  For example, *.bar.com would match a.bar.com and b.bar.com, but

it would not match a.x.bar.com nor would it match bar.com. If more than one identity of a given type is present in the certificate (e.g. more than one dNSName name), a match in any one of the set is considered acceptable.

If the hostname does not match the dNSName-based identity in the certificate per the above check, user-oriented clients SHOULD either notify the user (clients may give the user the opportunity to continue with the connection in any case) or terminate the connection and indicate that the server's identity is suspect. Automated clients SHOULD close the connection, returning and/or logging an error indicating that the server's identity is suspect.

Beyond the server identity checks described in this section, clients SHOULD be prepared to do further checking to ensure that the server is authorized to provide the service it is observed to provide. The client may need to make use of local policy information in making this determination.

3.1.7. Refresh of Server Capabilities Information

Upon TLS session establishment, the client SHOULD discard or refresh all information about the server it obtained prior to the initiation of the TLS negotiation and not obtained through secure mechanisms. This protects against man-in-the-middle attacks that may have altered any server capabilities information retrieved prior to TLS establishment.

The server may advertise different capabilities after TLS establishment. In particular, the value of supportedSASLMechanisms may be different after TLS has been negotiated (specifically, the EXTERNAL and PLAIN [PLAIN] mechanisms are likely to be listed only after a TLS negotiation has been performed).

3.2. Effects of TLS on a Client's Authorization Identity

This section describes the effects on a client's authorization identity brought about by establishing TLS on an LDAP connection. The default effects are described first, and next the facilities for client assertion of authorization identity are discussed including error conditions. Finally, the effects of closing the TLS connection are described.

Authorization identities and related concepts are described in Appendix B.

3.2.1. TLS Connection Establishment Effects

The decision to keep or invalidate the established state of the association (section 4.3) after TLS connection establishment is a matter of local server policy.

3.2.2. Client Assertion of Authorization Identity

After successfully establishing a TLS session, a client may request that its certificate exchanged during the TLS establishment be utilized to determine the authorization identity of the association. The client accomplishes this via an LDAP Bind request specifying a SASL mechanism of EXTERNAL [SASL] (section 10).

3.2.3. TLS Connection Closure Effects

The decision to keep or invalidate the established state of the association after TLS closure is a matter of local server policy.

3.3. TLS Ciphersuites

Several issues should be considered when selecting TLS ciphersuites that are appropriate for use in a given circumstance. These issues include the following:

- The ciphersuite's ability to provide adequate confidentiality protection for passwords and other data sent over the LDAP connection. Client and server implementers should recognize that some TLS ciphersuites provide no confidentiality protection while other ciphersuites that do provide confidentiality protection may be vulnerable to being cracked using brute force methods, especially in light of ever-increasing CPU speeds that reduce the time needed to successfully mount such attacks.

  Client and server implementers should carefully consider the value of the password or data being protected versus the level of confidentially protection provided by the ciphersuite to ensure that the level of protection afforded by the ciphersuite is appropriate.

- The ciphersuite's vulnerability (or lack thereof) to man-in-the-middle attacks. Ciphersuites vulnerable to man-in-the-middle attacks SHOULD NOT be used to protect passwords or sensitive data, unless the network configuration is such that the danger of a man-in-the-middle attack is tolerable.

3.3.1. TLS Ciphersuites Recommendations

[[TODO: Kurt will have someone from security to look at this and will propose how to handle discussion of specific TLS ciphersuites in this draft.]]

As of the writing of this document, the following recommendations regarding TLS ciphersuites are applicable. Because circumstances are constantly changing, this list must not be considered exhaustive, but is hoped that it will serve as a useful starting point for implementers.

The following ciphersuites defined in [TLS] MUST NOT be used for confidentiality protection of passwords or data:

TLS_NULL_WITH_NULL_NULL

TLS_RSA_WITH_NULL_MD5
TLS_RSA_WITH_NULL_SHA

The following ciphersuites defined in [TLS] can be cracked easily (less than a day of CPU time on a standard CPU in 2000) and are NOT RECOMMENDED for use in confidentiality protection of passwords or data:

TLS_RSA_EXPORT_WITH_RC4_40_MD5
TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5
TLS_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_DH_DSS_EXPORT_WITH_DES40_CBC_SHA
TLS_DH_RSA_EXPORT_WITH_DES40_CBC_SHA

TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_DH_anon_EXPORT_WITH_RC4_40_MD5
TLS_DH_anon_EXPORT_WITH_DES40_CBC_SHA

The following ciphersuites are vulnerable to man-in-the-middle attacks:

TLS_DH_anon_EXPORT_WITH_RC4_40_MD5
TLS_DH_anon_WITH_RC4_128_MD5
TLS_DH_anon_EXPORT_WITH_DES40_CBC_SHA
TLS_DH_anon_WITH_DES_CBC_SHA
TLS_DH_anon_WITH_3DES_EDE_CBC_SHA

4. Associations

Every LDAP connection has an associated authorization state referred to as the "association". The Bind operation defined in section 4.2 of [Protocol] and discussed further in section 5 below allows information to be exchanged between the client and server to change the authorization state of the association.

4.1. Anonymous Association on Unbound Connections

Prior to the successful completion of a Bind operation and during any subsequent authentication exchange, the association has an anonymous authorization state. Among other things this implies that the client need not send a Bind Request in the first PDU of the connection. The client may send any operation request prior to binding, and the server MUST treat it as if it had been performed after an anonymous bind operation (section 6). This association state is sometimes referred to as an implied anonymous bind.

4.2. Anonymous Association After Failed Bind

Upon receipt of a Bind request, the association is moved to an anonymous state and only upon successful completion of the authentication exchange (and the Bind operation) is the association moved to an authenticated state. Thus, a failed Bind operation produces an anonymous association.

4.3. Invalidated Associations

The server may move the association to an invalidated state at any time, e.g. if an established security layer between the client and server has unexpectedly failed or been compromised. While the connection has an invalid association, the server may reject any operation request other than Bind, Unbind, and StartTLS by responding with a resultCode of strongAuthRequired to indicate that the server requires stronger authentication before it will attempt to perform the requested operation. In practice, this means that the client needs to bind to(re)establish a suitably strong authorization

state on the association before the server will attempt to perform the requested operation.

5. Bind Operation

The Bind operation ([Protocol] section 4.2) allows authentication information to be exchanged between the client and server to establish a new authorization state on the association.

The Bind request typically specifies the desired authentication identity. Some Bind mechanisms also allow the client to specify the authorization identity. If the authorization identity is not specified, the server derives it from the authentication identity in an implementation-specific manner.

If the authorization identity is specified the server MUST verify that the client's authentication identity is permitted to assume (e.g. proxy for) the asserted authorization identity. The server MUST reject the Bind operation with an invalidCredentials resultCode in the Bind response if the client is not so authorized.

5.1. Simple Authentication Choice

The simple authentication choice of the Bind Operation provides three authentication mechanisms:

1. an anonymous authentication mechanism (section 6),

2. an unauthenticated authentication mechanism (section 7), and

3. a simple authentication mechanism using credentials consisting of a name (in the form of an LDAP distinguished name [LDAPDN]) and a password (section 8).

5.2. SASL Authentication Choice

The sasl authentication choice of the Bind Operation provides facilities for using any SASL mechanism (sections 9-11) including authentication mechanisms and other services (e.g. data security

services).

6. Anonymous Authentication Mechanism of Simple Bind

An LDAP client may use the anonymous authentication mechanism of the simple Bind choice to explicitly establish an anonymous association by sending a Bind request with a name value of zero length and with the simple authentication choice containing a password value of zero length.

7. Unauthenticated Authentication Mechanism of Simple Bind

An LDAP client may use the unauthenticated authentication mechanism of the simple Bind choice to establish an anonymous association by sending a Bind request with a name value, a distinguished name in

LDAP string form [LDAPDN], of non-zero length, and specifying the the simple authentication choice containing a password value of zero length.

Unauthenticated binds can have significant security issues (see section 12.3). Servers SHOULD by default reject unauthenticated bind requests with a resultCode of invalidCredentials, and clients may need to actively detect situations where they would unintentionally make an unauthenticated bind request.

8. Simple Authentication Mechanism of Simple Bind

An LDAP client may use the simple authentication mechanism of the simple Bind choice to establish an authenticated association by sending a Bind request with a name value, a distinguished name in LDAP string form [LDAPDN], and specifying the simple authentication choice containing an OCTET STRING password value of non-zero length.

Servers that map the DN sent in the bind request to a directory entry with an associated set of one or more passwords used with this mechanism, will compare the presented password to that set of passwords. The presented password is considered valid if it matches any member of this set.

If the DN is syntactically invalid, the server returns the invalidDNSyntax result code. If the DN is syntactically correct but not valid for purposes of authentication, or the password is not valid for the DN, or the server otherwise considers the credentials to be invalid, the server returns the invalidCredentials result code. The server is only to return the success result code when the credentials are valid and the server is willing to provide service to the entity these credentials identify.

Server behavior is undefined for bind requests specifying the simple authentication mechanism with a zero-length name value and a password value of non-zero length.

The simple authentication mechanism of simple bind is not suitable for authentication in environments where there is no network or transport layer confidentiality. LDAP implementations SHALL NOT support this mechanism unless they are capable of protecting it by

establishment of TLS (as discussed in section 3) or other suitable data confidentiality and data integrity protection(e.g. IPSec). LDAP implementations SHOULD support authentication with the "simple" authentication choice when the connection is protected against eavesdropping using TLS, as defined in section 3. LDAP implementations SHOULD NOT support authentication with the "simple" authentication choice unless the data on the connection is protected using TLS or other data confidentiality and data integrity protection.

9. SASL Protocol Profile

LDAP allows authentication via any SASL mechanism [SASL]. As LDAP includes native anonymous and simple (plain text) authentication methods, the ANONYMOUS [ANONYMOUS] and PLAIN [PLAIN] SASL mechanisms are typically not used with LDAP.

Each protocol that utilizes SASL services is required to supply certain information profiling the way they are exposed through the protocol ([SASL] section 5). This section explains how each of these profiling requirements are met by LDAP.

9.1. SASL Service Name for LDAP

The SASL service name for LDAP is "ldap", which has been registered with the IANA as a SASL service name.

9.2. SASL Authentication Initiation and Protocol Exchange

SASL authentication is initiated via an LDAP bind request ([Protocol] section 4.2) with the following parameters:

- The version is 3.
- The AuthenticationChoice is sasl.
- The mechanism element of the SaslCredentials sequence contains the value of the desired SASL mechanism.
- The optional credentials field of the SaslCredentials sequence may be used to provide an initial client response for mechanisms that are defined to have the client send data first (see [SASL] sections 5 and 5.1).

In general, a SASL authentication protocol exchange consists of a series of server challenges and client responses, the contents of which are specific to and defined by the SASL mechanism. Thus for some SASL authentication mechanisms, it may be necessary for the client to respond to one or more server challenges by invoking the BindRequest multiple times. A challenge is indicated by the server sending a BindResponse with the resultCode set to saslBindInProgress. This indicates that the server requires the client to send a new bind request with the same sasl mechanism to continue the authentication process.

To the LDAP protocol, these challenges and responses are opaque

binary tokens of arbitrary length. LDAP servers use the serverSaslCreds field, an OCTET STRING, in a bind response message to transmit each challenge. LDAP clients use the credentials field, an OCTET STRING, in the SaslCredentials sequence of a bind request message to transmit each response. Note that unlike some Internet protocols where SASL is used, LDAP is not text-based, thus no Base64 transformations are performed on these challenge and response values.

Clients sending a bind request with the sasl choice selected SHOULD send an zero-length value in the name field. Servers receiving a bind request with the sasl choice selected SHALL ignore any value in the name field.

A client may abort a SASL bind negotiation by sending a BindRequest with a different value in the mechanism field of SaslCredentials, or an AuthenticationChoice other than sasl.

If the client sends a BindRequest with the sasl mechanism field as an empty string, the server MUST return a BindResponse with authMethodNotSupported as the resultCode. This will allow clients to abort a negotiation if it wishes to try again with the same SASL mechanism.

The server indicates completion of the SASL challenge-response exchange by responding with a bind response in which the resultCode is either success, or an error indication.

The serverSaslCreds field in the BindResponse can be used to include an optional challenge with a success notification for mechanisms which are defined to have the server send additional data along with the indication of successful completion. If a server does not intend to send a challenge value in a BindResponse message, the server SHALL omit the serverSaslCreds field (rather than including the field with a zero-length value).

9.3. Octet Where Negotiated Security Mechanisms Take Effect

SASL security layers take effect following the transmission by the server and reception by the client of the final successful BindResponse in the exchange.

Once a SASL security layer providing data integrity or confidentiality services takes effect, the layer remains in effect until a new layer is installed (i.e. at the first octet following the final BindResponse of the bind operation that caused the new layer to take effect). Thus, an established SASL security layer is not affected by a failed or non-SASL Bind.

9.4. Determination of Supported SASL Mechanisms

Clients may determine the SASL mechanisms a server supports by reading the supportedSASLMechanisms attribute from the root DSE (DSA-Specific Entry) ([Models] section 5.1). The values of this attribute, if any, list the mechanisms the server supports in the current LDAP session state. LDAP servers SHOULD allow an

anonymously-bound client to retrieve the supportedSASLMechanisms attribute of the root DSE.

Because SASL mechanisms provide critical security functions, clients and servers should be configurable to specify what mechanisms are acceptable and allow only those mechanisms to be used. Both clients and servers must confirm that the negotiated security level meets their requirements before proceeding to use the connection.

9.5. Rules for Using SASL Security Layers

If a SASL security layer is negotiated, the client SHOULD discard information about the server it obtained prior to the initiation of the SASL negotiation and not obtained through secure mechanisms.

If a lower level security layer (such as TLS) is negotiated, any SASL security services SHALL be layered on top of such security layers regardless of the order of their negotiation. In all other respects, SASL security services and other security layers act independently, e.g. if both TLS and SASL security service are in effect then removing the SASL security service does not affect the continuing service of TLS and vice versa.

9.6 Support for Multiple Authentications

LDAP supports multiple SASL authentications as defined in [SASL] section 6.3.

10. SASL EXTERNAL Authentication Mechanism

A client can use the SASL EXTERNAL [SASL] mechanism to request the LDAP server to authenticate and establish a resulting authorization identity using security credentials exchanged by a lower security layer (such as by TLS authentication or IP-level security [RFC2401]).

The authorization identity used to determine the state of the association is derived from the security credentials in an implementation-specific manner. If the client's authentication credentials have not been established at a lower security layer, the SASL EXTERNAL bind MUST fail with a resultCode of inappropriateAuthentication. Although this situation has the effect of leaving the association in an anonymous state (section 5), the state of any established security layer is unaffected.

A client may either implicitly request that its authorization identity be derived from its authentication credentials exchanged at a lower security layer or it may explicitly provide an authorization identity and assert that it be used in combination with those authentication credentials. The former is known as an implicit assertion, and the latter as an explicit assertion.

10.1. Implicit Assertion

An implicit authorization identity assertion is performed by invoking a Bind request of the SASL form using the EXTERNAL mechanism name that does not include the optional credentials octet string (found within the SaslCredentials sequence in the Bind Request). The server will derive the client's authorization identity from the authentication identity supplied by the security layer (e.g., a public key certificate used during TLS establishment) according to local policy. The underlying mechanics of how this is accomplished are implementation specific.

10.2. Explicit Assertion

An explicit authorization identity assertion is performed by invoking a Bind request of the SASL form using the EXTERNAL mechanism name that includes the credentials octet string. This string MUST be constructed as documented in section 10.4.

10.3. SASL Authorization Identity

When the EXTERNAL SASL mechanism is being negotiated, if the SaslCredentials credentials field is present, it contains an authorization identity. Other mechanisms define the location of the authorization identity in the credentials field. In either case, the authorization identity is represented in the authzId form described below.

10.4. SASL Authorization Identity Syntax

The authorization identity is a string of UTF-8 [RFC3629] encoded [Unicode] characters corresponding to the following ABNF [RFC2234] grammar:

```
authzId ::= dnAuthzId / uAuthzId

DNCOLON  ::= %x64 %x6e %x3a ; "dn:"
UCOLON ::= %x75 %x3a ; "u:"

; distinguished-name-based authz id.
dnAuthzId ::= DNCOLON distinguishedName

; unspecified authorization id, UTF-8 encoded.
uAuthzId ::= UCOLON userid
userid ::= *UTF8    ; syntax unspecified
```

where the <distinguishedName> production is defined in section 3 of [LDAPDN] and <UTF8> production is defined in section 1.3 of [Models].

In order to support additional specific authorization identity forms, future updates to this specification may add new choices supporting other forms of the authzId production.

The dnAuthzId choice is used to assert authorization identities in the form of a distinguished name to be matched in accordance with the distinguishedNameMatch matching rule [Syntaxes]. The decision to allow or disallow an authentication identity to have access to the requested authorization identity is a matter of local policy ([SASL] section 4.2). For this reason there is no requirement that the asserted dn be that of an entry in the directory.

The uAuthzId choice allows clients to assert an authorization identity that is not in distinguished name form. The format of userid is defined as only a sequence of UTF-8 [RFC3629] encoded [Unicode] characters, and any further interpretation is a local matter. To compare uAuthzID values, each uAuthzID value MUST be

prepared using [SASLPrep] and then the two values are compared octet-wise.

For example, the userid could identify a user of a specific directory service, be a login name, or be an email address. A uAuthzId SHOULD NOT be assumed to be globally unique.

11. SASL DIGEST-MD5 Authentication Mechanism

LDAP servers that implement any authentication method or mechanism other than simple anonymous bind MUST implement the SASL DIGEST-MD5 mechanism [DIGEST-MD5]. This provides client authentication with protection against passive eavesdropping attacks but does not provide protection against man-in-the-middle attacks. DIGEST-MD5 also provides data integrity and data confidentiality capabilities.

Support for subsequent authentication ([DIGEST-MD5] section 2.2) is OPTIONAL in clients and servers.

Implementers must take care to ensure that they maintain the semantics of the DIGEST-MD5 specification even when handling data that has different semantics in the LDAP protocol.
For example, the SASL DIGEST-MD5 authentication mechanism utilizes realm and username values ([DIGEST-MD5] section 2.1) which are syntactically simple strings and semantically simple realm and username values. These values are not LDAP DNs, and there is no requirement that they be represented or treated as such. Username and realm values that look like LDAP DNs in form, e.g. <cn=bob, dc=example,dc=com>, are syntactically allowed, however DIGEST-MD5 treats them as simple strings for comparison purposes. To illustrate further, the two DNs <cn=Bob,dc=example,dc=com> (upper case "B") and <cn=bob,dc=example,dc=com> (lower case "b") are equivalent when being compared semantically as LDAP DNs because the cn attribute is defined to be case insensitive, however the two values are not equivalent if they represent username values in DIGEST-MD5 because [SASLPrep] semantics are used by DIGEST-MD5.

12. Security Considerations

Security issues are discussed throughout this document. The unsurprising conclusion is that security is an integral and

necessary part of LDAP. This section discusses a number of LDAP-related security considerations.

12.1. General LDAP Security Considerations

LDAP itself provides no security or protection from accessing or updating the directory by other means than through the LDAP protocol, e.g. from inspection by database administrators. Access control SHOULD always be applied when reading sensitive information or updating directory information.

Servers can minimize denial of service attacks by providing the ability to configure and enforce administrative limits on operations, timing out idle connections and returning the unwillingToPerform resultCode rather than performing computationally expensive operations requested by unauthorized clients.

A connection on which the client has not established connection integrity and privacy services (e.g via StartTLS, IPSec or a suitable SASL mechanism) is subject to man-in-the-middle attacks to view and modify information in transit. Client and server implementors SHOULD take measures to protect confidential data from these attacks by using data protection services as discussed in this document.

12.1.1. Password-related Security Considerations

LDAP allows multi-valued password attributes. In systems where entries are expected to have one and only one password, administrative controls should be provided to enforce this behavior.

The use of clear text passwords and other unprotected authentication credentials is strongly discouraged over open networks when the underlying transport service cannot guarantee confidentiality.

The transmission of passwords in the clear--typically for authentication or modification--poses a significant security risk. This risk can be avoided by using SASL authentication [SASL] mechanisms that do not transmit passwords in the clear or by negotiating transport or session layer data confidentiality services before transmitting password values.

To mitigate the security risks associated with the transfer of passwords, a server implementation that supports any password-based authentication mechanism that transmits passwords in the clear MUST support a policy mechanism that at the time of authentication or password modification, requires:

A StartTLS encryption layer has been successfully negotiated.

OR

Some other data confidentiality mechanism that protects the password value from snooping has been provided.

OR

The server returns a resultCode of confidentialityRequired for the operation (i.e. simple bind with password value, SASL bind transmitting a password value in the clear, add or modify including a userPassword value, etc.), even if the password value is correct.

12.2. StartTLS Security Considerations

All security gained via use of the StartTLS operation is gained by the use of TLS itself. The StartTLS operation, on its own, does not provide any additional security.

The level of security provided though the use of TLS depends directly on both the quality of the TLS implementation used and the style of usage of that implementation. Additionally, a man-in-the-middle attacker can remove the StartTLS extended operation from the supportedExtension attribute of the root DSE. Both parties SHOULD independently ascertain and consent to the security level achieved once TLS is established and before beginning use of the TLS connection. For example, the security level of the TLS connection might have been negotiated down to plaintext.

Clients SHOULD by default either warn the user when the security level achieved does not provide an acceptable level of data confidentiality and/or data integrity protection, or be configured to refuse to proceed without an acceptable level of security.

Server implementors SHOULD allow server administrators to elect whether and when data confidentiality and integrity are required, as well as elect whether authentication of the client during the TLS handshake is required.

Implementers should be aware of and understand TLS security considerations as discussed in the TLS specification [TLS].

12.3. Unauthenticated Mechanism Security Considerations

Operational experience shows that clients can (and frequently do) misuse the unauthenticated authentication mechanism of simple bind (see section 7). For example, a client program might make a decision to grant access to non-directory information on the basis of completing a successful bind operation. LDAP server implementations may return a success response to an unauthenticated bind request thus leaving the client with the impression that the server has successfully authenticated the identity represented by the user name, when in effect, an anonymous association has been established. Clients that use the results from a simple bind operation to make authorization decisions should actively detect unauthenticated bind requests (by verifying that the supplied

password is not empty) and react appropriately.

12.4. Simple Mechanism Security Considerations

The simple authentication mechanism of simple bind discloses the password to the server, which is an inherent security risk. There are other mechanisms such as DIGEST-MD5 that do not disclose password to server.

12.5. SASL DIGEST-MD5 Mechanism Security Considerations

The SASL DIGEST-MD5 mechanism is prone to the qop substitution attack, as discussed in 3.6 of [DIGEST-MD5]. The qop substitution attack can be mitigated (as discussed in 3.6 of [DIGEST-MD5]).

The SASL DIGEST-MD5 mechanism [DIGEST-MD5] provides client authentication with protection against passive eavesdropping attacks but does not provide protection against man-in-the-middle attacks.

Implementers should be aware of and understand DIGEST-MD5 security considerations as discussed in the DIGEST-MD5 specification [DIGEST-MD5].

12.6. Related Security Considerations

Additional security considerations relating to the various authentication methods and mechanisms discussed in this document apply and can be found in [SASL], [SASLPrep], [StringPrep] and [RFC3629].

13. IANA Considerations

The following IANA considerations apply to this document:

It is requested that the IANA update the LDAP Protocol Mechanism registry to indicate that this document and [Protocol] provide the definitive technical specification for the StartTLS (1.3.6.1.4.1.1466.20037) extended operation.

[[TODO: add any missing IANA Considerations.]]

Acknowledgments

This document combines information originally contained in RFC 2829 and RFC 2830. The editor acknowledges the work of Harald Tveit Alvestrand, Jeff Hodges, Tim Howes, Steve Kille, RL "Bob" Morgan , and Mark Wahl, each of whom authored one or more of these documents.

This document is based upon input of the IETF LDAP Revision working group. The contributions and suggestions made by its members in

shaping the contents and technical accuracy of this document is greatly appreciated.

Normative References

[[Note to the RFC Editor: please replace the citation tags used in referencing Internet-Drafts with tags of the form RFCnnnn.]]

[RFC2234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[DIGEST-MD5] Leach, P. C. Newman, and A. Melnikov, "Using Digest Authentication as a SASL Mechanism", draft-ietf-sasl-rfc2831bis-xx.txt, a work in progress.

[RFC2119] Bradner, S., "Key Words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[LDAPDN] Zeilenga, Kurt D. (editor), "LDAP: String Representation of Distinguished Names", draft-ietf-ldapbis-dn-xx.txt, a work in progress.

[Matching] Hoffman, Paul and Steve Hanna, "Matching Text Strings in PKIX Certificates", draft-hoffman-pkix-stringmatch-xx.txt, a work in progress.

[Models] Zeilenga, Kurt D. (editor), "LDAP: Directory Information Models", draft-ietf-ldapbis-models-xx.txt, a work in progress.

[Protocol] Sermersheim, J., "LDAP: The Protocol", draft-ietf-ldapbis-protocol-xx.txt, a work in progress.

[Roadmap] K. Zeilenga, "LDAP: Technical Specification Road Map", draft-ietf-ldapbis-roadmap-xx.txt, a work in progress.

[SASL] Melnikov, A. (editor), "Simple Authentication and Security Layer (SASL)", draft-ietf-sasl-rfc2222bis-xx.txt, a work in progress.

[SASLPrep] Zeilenga, K., "Stringprep profile for user names and passwords", draft-ietf-sasl-saslprep-xx.txt, (a work in progress).

[StringPrep] M. Blanchet, "Preparation of Internationalized Strings ('stringprep')", draft-hoffman-rfc3454bis-xx.txt, a work in progress.

[Syntaxes] Legg, S. (editor), "LDAP: Syntaxes and Matching Rules", draft-ietf-ldapbis-syntaxes-xx.txt, a work in progress.

[TLS] Dierks, T. and C. Allen. "The TLS Protocol Version 1.1", draft-ietf-tls-rfc2246-bis-xx.txt, a work in progress.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", RFC 3629, STD 63, November 2003.

[Unicode] The Unicode Consortium, "The Unicode Standard, Version 3.2.0" is defined by "The Unicode Standard, Version 3.0" (Reading, MA, Addison-Wesley, 2000. ISBN 0-201-61633-5), as amended by the "Unicode Standard Annex #27: Unicode 3.1" (http://www.unicode.org/reports/tr27/) and by the "Unicode Standard Annex #28: Unicode 3.2" (http://www.unicode.org/reports/tr28/).

Informative References

[ANONYMOUS] Zeilenga, K.,"Anonymous SASL Mechanism", draft-zeilenga-sasl-anon-xx.txt, a work in progress.

[RFC2828] Shirey, R., "Internet Security Glossary", RFC 2828, May 2000.

[PLAIN] Zeilenga, K.,"Plain SASL Mechanism", draft-zeilenga-sasl-plain-xx.txt, a work in progress.

[RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

Author's Address

Roger Harrison
Novell, Inc.
1800 S. Novell Place
Provo, UT 84606
USA
+1 801 861 2642
roger_harrison@novell.com

Appendix A. Association State Transition Tables

This section provides a state transition table to represent a state diagram for the various authentication states through which an association may pass during the course of its existence and the actions that cause these changes in state.

This section is based entirely on information found in this document and other documents that are part of the LDAP Technical Specification [Roadmap]. As such, it is strictly informational in nature.

A.1. Association States

The following table lists the valid association states and provides a description of each state. The ID for each state is used in the state transition table in section A.4.

```
ID State Description
-- ------------------------------------------------------------
S1 Anonymous
       no Authentication ID is associated with the LDAP connection
       no Authorization ID is in force
S2 Authenticated
       Authentication ID = I
       Authorization ID = X
S3 Authenticated SASL EXTERNAL, implicit authorization ID
       Authentication ID = J
       Authorization ID = Y
```

S4 Authenticated SASL EXTERNAL, explicit authorization ID Z
Authentication ID = J
Authorization ID = Z
S5 Invalidated

A.2. Actions that Affect Association State

The following table lists the actions that can affect the authentication and authorization state of an association. The ID for each action is used in the state transition table in section A.4.

ID	Action
A1	Client bind request fails
A2	Client successfully performs anonymous simple bind or unauthenticated simple bind
A3	Client successfully performs simple bind with name and password OR SASL bind with any mechanism except EXTERNAL using an authentication ID = I that maps to authorization ID X
A4	Client Binds SASL EXTERNAL with implicit assertion of authorization ID (section 9.1). The current authentication ID maps to authorization ID = Y.
A5	Client Binds SASL EXTERNAL with explicit assertion of authorization ID = Z (section 9.2).
A6	Client StartTLS request fails
A7	Client StartTLS request succeeds
A8	Client or Server: graceful TLS removal
A9	Server decides to invalidate current association state

A.3. Decisions Used in Making Association State Changes

Certain changes in the authentication and authorization state of an association are only allowed if the server can affirmatively answer a question. These questions are applied as part of the criteria for allowing or disallowing a state transition in the state transition table in section A.4.

ID	Decision Question
D1	Are lower-layer credentials available?
D2	Can lower-layer credentials for Auth ID "K" be mapped to asserted AuthZID "L"?

A.4. Association State Transition Table

The Association table below lists the the actions that could affect the authorization state of an association and the resulting state of an association after a given action occurs.

S1, the initial state for the state machine described in this table, is the association state when an LDAP connection is initially established.

Action	Next State	Comment
A1	S1	Section 4
A2	S1	Sections 6 & 7
A3	S2	Sections 8, 9
A4, D1=no	S1	Failed bind, section 10.1
A4, D1=yes	S3	
A5, D1=no	S1	Failed bind, section 10.2
A5, D1=yes, D2=no	S1	Failed bind, section 10.2
A5, D1=yes, D2=yes	S4	
A6	no change*	[Protocol] section 4.14.2.2
A7	no change*	[Protocol] section 4.14.2.1
A8	S1	[Protocol] section 4.14.3.1
A9	S5	

* The server may invalidate the association after TLS establishment or closure (section 3.2).

Appendix B. Authentication and Authorization Concepts

This appendix defines basic terms, concepts, and interrelationships regarding authentication, authorization, credentials, and identity. These concepts are used in describing how various security approaches are utilized in client authentication and authorization.

B.1. Access Control Policy

An access control policy is a set of rules defining the protection of resources, generally in terms of the capabilities of persons or other entities accessing those resources. Security objects and mechanisms, such as those described here, enable the expression of access control policies and their enforcement.

B.2. Access Control Factors

A request, when it is being processed by a server, may be associated with a wide variety of security-related factors (section 4.2 of [Protocol]). The server uses these factors to determine whether and how to process the request. These are called access control factors (ACFs). They might include source IP address, encryption strength, the type of operation being requested, time of day, etc. Some factors may be specific to the request itself, others may be associated with the connection via which the request is transmitted, others (e.g. time of day) may be "environmental".

Access control policies are expressed in terms of access control factors. E.g., a request having ACFs i,j,k can perform operation Y

on resource Z. The set of ACFs that a server makes available for such expressions is implementation-specific.

B.3. Authentication, Credentials, Identity

Authentication credentials are the evidence supplied by one party to another, asserting the identity of the supplying party (e.g. a user) who is attempting to establish a new association state with the other party (typically a server). Authentication is the process of generating, transmitting, and verifying these credentials and thus the identity they assert. An authentication identity is the name presented in a credential.

There are many forms of authentication credentials -- the form used depends upon the particular authentication mechanism negotiated by the parties. For example: X.509 certificates, Kerberos tickets, simple identity and password pairs. Note that an authentication mechanism may constrain the form of authentication identities used with it.

B.4. Authorization Identity

An authorization identity is one kind of access control factor. It is the name of the user or other entity that requests that operations be performed. Access control policies are often expressed in terms of authorization identities; e.g., entity X can perform operation Y on resource Z.

The authorization identity bound to an association is often exactly the same as the authentication identity presented by the client, but it may be different. SASL allows clients to specify an authorization identity distinct from the authentication identity asserted by the client's credentials. This permits agents such as proxy servers to authenticate using their own credentials, yet request the access privileges of the identity for which they are proxying [SASL]. Also, the form of authentication identity supplied by a service like TLS may not correspond to the authorization identities used to express a server's access control policy, requiring a server-specific mapping to be done. The method by which a server composes and validates an authorization identity from the authentication credentials supplied by a client is performed in an implementation-specific manner.

Appendix C. RFC 2829 Change History

This appendix lists the changes made to the text of RFC 2829 in preparing this document.

C.0. General Editorial Changes

Version -00

- Changed other instances of the term LDAP to LDAP where v3 of the protocol is implied. Also made all references to LDAP use the same wording.

- Miscellaneous grammatical changes to improve readability.

- Made capitalization in section headings consistent.

Version -01

- Changed title to reflect inclusion of material from RFC 2830 and 2251.

C.1. Changes to Section 1

Version -01

- Moved conventions used in document to a separate section.

C.2. Changes to Section 2

Version -01

- Moved section to an appendix.

C.3. Changes to Section 3

Version -01

- Moved section to an appendix.

C.4 Changes to Section 4

Version -00

- Changed "Distinguished Name" to "LDAP distinguished name".

C.5. Changes to Section 5

Version -00

- Added the following sentence: "Servers SHOULD NOT allow clients with anonymous authentication to modify directory entries or access sensitive information in directory entries."

C.5.1. Changes to Section 5.1

Version -00

- Replaced the text describing the procedure for performing an anonymous bind (protocol) with a reference to section 4.2 of RFC 2251 (the protocol spec).

Version -01

- Brought text describing procedure for performing an anonymous bind from section 4.2 of RFC 2251 bis. This text will be removed from the draft standard version of that document.

C.6. Changes to Section 6.

Version -00

Reorganized text in section 6.1 as follows:

1. Added a new section (6.1) titled "Simple Authentication" and moved one of two introductory paragraphs for section 6 into section 6.1. Added sentences to the paragraph indicating:

   a. simple authentication is not suitable for environments where confidentiality is not available.

   b. LDAP implementations SHOULD NOT support simple authentication unless confidentiality and data integrity mechanisms are in force.

2. Moved first paragraph of section 6 (beginning with "LDAP implementations MUST support authentication with a password...") to section on Digest Authentication (Now section 6.2).

C.6.1. Changes to Section 6.1.

Version -00 Renamed section to 6.2

- Added sentence from original section 6 indicating that the DIGEST-MD5 SASL mechanism is required for all conforming LDAP implementations

C.6.2. Changes to Section 6.2

Version -00

- Renamed section to 6.3

- Reworded first paragraph to remove reference to user and the

userPassword password attribute Made the first paragraph more general by simply saying that if a directory supports simple authentication that the simple bind operation MAY performed following negotiation of a TLS ciphersuite that supports confidentiality.

- Replaced "the name of the user's entry" with "a DN" since not all bind operations are performed on behalf of a "user."

- Added Section 6.3.1 heading just prior to paragraph 5.

- Paragraph 5: replaced "The server" with "DSAs that map the DN sent in the bind request to a directory entry with a userPassword attribute."

C.6.3. Changes to section 6.3.

Version -00

- Renamed to section 6.4.

C.7. Changes to section 7.

none

C.7.1. Changes to section 7.1.

Version -00

- Clarified the entity issuing a certificate by moving the phrase "to have issued the certificate" immediately after "Certification Authority."

C.8. Changes to section 8.

Version -00

- Removed the first paragraph because simple authentication is covered explicitly in section 6.

- Added section 8.1. heading just prior to second paragraph.

- Added section 8.2. heading just prior to third paragraph.

- Added section 8.3. heading just prior to fourth paragraph.

Version -01

- Moved entire section 8 of RFC 2829 into section 3.4 (Using SASL for Other Security Services) to bring material on SASL mechanisms together into one location.

C.9. Changes to section 9.

Version -00

- Paragraph 2: changed "EXTERNAL mechanism" to "EXTERNAL SASL mechanism."

- Added section 9.1. heading.

- Modified a comment in the ABNF from "unspecified userid" to "unspecified authz id".

- Deleted sentence, "A utf8string is defined to be the UTF-8 encoding of one or more ISO 10646 characters," because it is redundant.

- Added section 9.1.1. heading.

- Added section 9.1.2. heading.

Version -01

- Moved entire section 9 to become section 3.5 so that it would be with other SASL material.

C.10. Changes to Section 10.

Version -00

- Updated reference to cracking from a week of CPU time in 1997 to be a day of CPU time in 2000.

- Added text: "These ciphersuites are NOT RECOMMENDED for use... and server implementers SHOULD" to sentence just prior the second list of ciphersuites.

- Added text: "and MAY support other ciphersuites offering equivalent or better protection," to the last paragraph of the section.

C.11. Changes to Section 11.

Version -01

- Moved to section 3.6 to be with other SASL material.

C.12. Changes to Section 12.

Version -00

- Inserted new section 12 that specifies when SASL protections begin following SASL negotiation, etc. The original section 12 is renumbered to become section 13.

Version -01

- Moved to section 3.7 to be with other SASL material.

C.13. Changes to Section 13 (original section 12).

None

Appendix D. RFC 2830 Change History

This appendix lists the changes made to the text of RFC 2830 in preparing this document.

D.0. General Editorial Changes

- Material showing the PDUs for the StartTLS response was broken out into a new section.

- The wording of the definition of the StartTLS request and StartTLS response was changed to make them parallel. NO changes were made to the ASN.1 definition or the associated values of the parameters.

- A separate section heading for graceful TLS closure was added for parallelism with section on abrupt TLS closure.

Appendix E. RFC 2251 Change History

This appendix lists the changes made to the text of RFC 2251 in preparing this document.

E.0. General Editorial Changes

- All material from section 4.2 of RFC 2251 was moved into this document.

- A new section was created for the Bind Request

- Section 4.2.1 of RFC 2251 (Sequencing Bind Request) was moved after the section on the Bind Response for parallelism with the presentation of the StartTLS operations. The section was also subdivided to explicitly call out the various effects being described within it.

- All SASL profile information from RFC 2829 was brought within the discussion of the Bind operation (primarily sections 4.4 - 4.7).

Appendix F. Change History to Combined Document

F.1. Changes for draft-ldap-bis-authmeth-02

General

- Added references to other LDAP standard documents, to sections within the document, and fixed broken references.

- General editorial changes--punctuation, spelling, formatting, etc.

Section 1.

- Added glossary of terms and added sub-section headings

Section 2.

- Clarified security mechanisms 3, 4, & 5 and brought language in line with IETF security glossary.

Section 3.

- Brought language in requirement (3) in line with security glossary.

- Clarified that information fetched prior to initiation of TLS negotiation must be discarded

-Clarified that information fetched prior to initiation of SASL negotiation must be discarded

- Rewrote paragraph on SASL negotiation requirements to clarify intent

Section 4.4.

- Added stipulation that sasl choice allows for any SASL mechanism not prohibited by this document. (Resolved conflict between this statement and one that prohibited use of ANONYMOUS and PLAIN SASL mechanisms.)

Section 5.3.6

- Added a.x.bar.com to wildcard matching example on hostname check.

Section 6

- Added Association State Transition Tables to show the various states through which an association may pass along with the actions and decisions required to traverse from state to state.

Appendix A

- Brought security terminology in line with IETF security glossary throughout the appendix.

F.2. Changes for draft-ldapbis-authmeth-03

General

- Added introductory notes and changed title of document and references to conform to WG chair suggestions for the overall technical specification.

- Several issues--H.13, H.14, H.16, H.17--were resolved without requiring changes to the document.

Section 3

- Removed reference to /etc/passwd file and associated text.

Section 4

- Removed sections 4.1, 4.2 and parts of section 4.3. This information was being duplicated in the protocol specification and will now reside there permanently.

Section 4.2

- changed words, "not recommended" to "strongly discouraged"

Section 4.3

- Based on ldapbis WG discussion at IETF52 two sentences were added indicating that clients SHOULD NOT send a DN value when binding with the sasl choice and servers SHALL ignore any value received in this circumstance.
-

Section 8.3.1

- Generalized the language of this section to not refer to any specific password attribute or to refer to the directory entry as a "user" entry.

Section 11

- Added security consideration regarding misuse of unauthenticated access.

- Added security consideration requiring access control to be applied only to authenticated users and recommending it be applied when reading sensitive information or updating directory information.

F.3. Changes for draft-ldapbis-authmeth-04

General

- Changed references to use [RFCnnnn] format wherever possible. (References to works in progress still use [name] format.)
- Various edits to correct typos and bring field names, etc. in line with specification in [Protocol] draft.

- Several issues--H.13, H.14, H.16, H.17--were resolved without

requiring changes to the document.

Section 4.4.1.

- Changed ABNF grammar to use productions that are like those in the model draft.

Section 5

- Removed sections 5.1, 5.2, and 5.4 that will be added to [Protocol]. Renumbered sections to accommodate this change.
-

Section 6

- Reviewed Association State table for completeness and accuracy. Renumbered actions A3, , and A5 to be A5, A3, and A4 respectively. Re-ordered several lines in the table to ensure that actions are in ascending order (makes analyzing the table much more logical). Added action A2 to several states where it was missing and valid. Added actions A7 and A8 placeholders to states S1, S2, S4 and S5 pending resolution of issue H.28.

Section 11

- Modified security consideration (originally added in -03) requiring access control to be applied only to authenticated users. This seems nonsensical because anonymous users may have access control applied to limit permissible actions.
-

Section 13

- Verified all normative references and moved informative references to a new section 14.

F.4. Changes for draft-ldapbis-authmeth-05

General

- General editory changes to fix punctuation, spelling, line length issues, etc.
- Verified and updated intra- and inter-document references throughout.
- Document-wide review for proper usage of RFC 2119 keywords with several changes to correct improper usage.

Abstract

- Updated to match current contents of documents. This was needed due to movement of material on Bind and StartTLS operations to [Protocol] in this revision.

Section 3.

- Renamed section to "Rationale for LDAP Security Mechanisms" and removed text that did not support this theme. Part of the

motivation for this change was to remove the implication of the previous section title, "Required Security Mechanisms", and other text found in the section that everything in the section was a requirement

- Information from several removed paragraphs that describe deployment scenarios will be added Appendix A in the next revision of the draft.

- Paragraph beginning, " If TLS is negotiated, the client MUST discard all information..." was moved to section 5.1.7 and integrated with related material there.

- Paragraph beginning, "If a SASL security layer is negotiated..." was moved to section 4.2

Section 4.l.

- Changed wording of first paragraph to clarify meaning.

Section 4.2.

- Added paragraph from section 3 of -04 beginning, "If a SASL security layer is negotiated..."

Section 4.3.3.

- Renamed to "Other SASL Mechanisms" and completely rewrote the section (one sentence) to generalize the treatment of SASL mechanisms not explicitly mentioned in this document.

Section 4.4.1.

- Added paragraph beginning, "The dnAuthzID choice allows client applications..." to clarify whether DN form authorization identities have to also have a corresponding directory entry. This change was based on editor's perception of WG consensus.

- Made minor clarifying edits in the paragraph beginning, "The uAuthzID choice allows for compatibility..."

Section 5.1.1.

- Made minor clarifying edits in the last paragraph of the section.

Section 5.1.7.

- Wording from section 3 paragraph beginning " If TLS is negotiated, the client MUST discard all information..." was moved to this section and integrated with existing text.

Section 5.2.

- Changed usage of "TLS connection" to "TLS session" throughout.

- Removed empty section 5.2.1 and renumbered sections it had previously contained.

Section 8.

- Added introductory paragraph at beginning of section.

Section 8.1.

- Changed term "data privacy" to "data confidentiality" to be consistent with usage in rest of document.

Section 8.2.

- Changed first paragraph to require implementations that implement *password-based* authentication to implement and support DIGEST-MD5 SASL authentication.

Section 11.

- First paragraph: changed "session encryption" to "session confidentiality protection" to be consistent with usage in rest of document.

Appendix B.

- Began changes to incorporate information on deployment scenarios removed from section 3.

F.5. Changes for draft-ldapbis-authmeth-06

General

- Combined Section 2 (Introduction) and Section 3 (Motivation) and moved Introduction to section 1. All following sections numbers were decremented by one as result.

- Edits to fix typos, I-D nits, etc.

- Opened several new issues in Appendix G based on feedback from WG. Some of these have been resolved. Others require further discussion.

Section 1

- Added additional example of spoofing under threat (7).

Section 2.1

- Changed definition of "association" and added terms,

"connection" and "TLS connection" to bring usage in line with [Protocol].

Section 4.1.6

- Clarified sentence stating that the client MUST NOT use derived forms of DNS names.

Section 5.1

- Began edits to association state table to clarify meaning of various states and actions.

- Added action A9 to cover abandoned bind operation and added appropriate transitions to the state transition table to accommodate it.

Section 7.2

- Replaced first paragraph to clarify that the "DIGEST-MD5" SASL mechanism is required to implement.

Section 9

- Rewrote the section to make the advice more applicable over the long term, i.e. more "timeless." The intent of content in the original section was preserved.

Section 10

- Added a clarifying example to the consideration regarding misuse of unauthenticated access.

F.6. Changes for draft-ldapbis-authmeth-07

General

- Updated external and internal references to accommodate changes in recent drafts.

- Opened several new issues in Appendix G based on feedback from WG. Some of these have been resolved. Others require further discussion.

Section 3

- Rewrote much of section 3.3 to meet the SASL profile requirements of draft-ietf-sasl-rfc2222bis-xx.txt section 5.

- Changed treatement of SASL ANONYMOUS and PLAIN mechanisms to bring in line with WG consensus.

Section 4

- Note to implementers in section 4.1.1 based on operational experience.

- Clarification on client continuing by performing a StartTLS with TLS already established in section 4.1.4.

- Moved verification of mapping of client's authentication ID to asserted authorization ID to apply only to explicit assertion. The local policy in place for implicit assertion is adequate.

Section 7

- Removed most of section 7.2 as the information is now covered adequately via the new SASL profile in section 3.3. Added note to implementors regarding the treatment of username and realm values in DIGEST-MD5.

- Section 7.3. Minor clarifications in wording.

- Section 7.3.1. Clarification that a match of the presented value to any member of the set of stored passwords constitutes a successful authentication.

F.7. Changes for draft-ldapbis-authmeth-08

General

- Changed usage from LDAPv3 to LDAP for usage consistency across LDAP technical specification.

- Fixed a number of usage nits for consistency and to bring doc in conformance with publication guidelines.

Abstract

- Significant cleanup and rewording of abstract based on WG feedback.

Section 2.1

- New definition of user.

Section 3

- Added 1.5 sentences at end of introductory paragraph indicating the effect of the Bind op on the association.

Section 3.1

- Retitled section and clarified wording

Section 3.2

- Clarified that simple authentication choice provides three types of authentication: anonymous, unauthenticated, and simple password.

Section 3.3.3

- New wording clarifying when negotiated security mechanisms take effect.

Section 3.3.5

- Changed requirement to discard information about server fetched prior to SASL negotiation from MUST to SHOULD to allow for information obtained through secure mechanisms.

Section 3.3.6

- Simplified wording of first paragraph based on suggestion from WG.

Section 3.4

- Minor clarifications in wording.

Section 3.4.1

- Minor clarifications in wording in first sentence.
- Explicitly called out that the DN value in the dnAuthzID form is to be matched using DN matching rules.
- Called out that the uAuthzID MUST be prepared using SASLprep rules before being compared.
- Clarified requirement on assuming global uniqueness by changing a "generally... MUST" wording to "SHOULD".

Section 4.1.1

- Simplified wording describing conditions when StartTLS cannot be sent.
- Simplified wording in note to implementers regarding race condition with outstanding LDAP operations on connection.

Section 4.1.5

- Removed section and moved relevant text to section 4.2.2.

Section 4.1.6

- Renumbered to 4.1.5.
- Updated server identity check rules for server's name based on WG list discussion.

Section 4.1.7

- Renumbered to 4.1.6
- Changed requirement to discard information about server fetched

prior to TLS negotion from MUST to SHOULD to allow for information obtained through secure mechanisms.

Section 6.1

- Clarified wording.
- Added definition of anonymous and unauthenticated binds.

Section 10

- Added security consideration (moved from elsewhere) discouraging use of cleartext passwords on unprotected communication channels.

Section 11

- Added an IANA consideration to update GSSAPI service name registry to point to [Roadmap] and [Authmeth]

F.8. Changes for draft-ldapbis-authmeth-09

General

- Updated section references within document
- Changed reference tags to match other docs in LDAP TS
- Used non-quoted names for all SASL mechanisms

Abstract

- Inspected keyword usage and removed several improper usages.

- Removed sentence saying DIGEST-MD5 is LDAP's mandatory-to-implement mechanism. This is covered elsewhere in document.

- Moved section 5, authentication state table, of -08 draft to section 8 of -09 and completely rewrote it.

Section 1

- Reworded sentence beginning, "It is also desirable to allow authentication methods to carry identities based on existing, non-LDAP DN-forms..."
- Clarified relationship of this document to other documents in the LDAP TS.

Section 3.3.5

- Removed paragraph beginning,"If the client is configured to support multiple SASL mechanisms..." because the actions specified in the paragraph do not provide the protections indicated. Added a new paragraph indicating that clients and server should allow specification of acceptable mechanisms and only allow those mechanisms to be used.

- Clarified independent behavior when TLS and SASL security layers are both in force (e.g. one being removed doesn't affect the other).

Section 3.3.6

- Moved most of section 4.2.2, Client Assertion of Authorization Identity, to sections 3.3.6, 3.3.6.1, and 3.3.6.2.

Section 3.3.6.4

- Moved some normative comments into text body.

Section 4.1.2

- Non success resultCode values are valid if server is *unwilling* or unable to negotiate TLS.

Section 4.2.1

- Rewrote entire section based on WG feedback.

Section 4.2.2

- Moved most of this section to 3.3.6 for better document flow.

Section 4.2.3

- Rewrote entire section based on WG feedback.

Section 5.1

- Moved imperative language regarding unauthenticated access from security considerations to here.

Section 6

- Added several paragraphs regarding the risks of transmitting passwords in the clear and requiring server implementations to provide a specific configuration that reduces these risks.

Section 6.2

- Added sentence describing protections provided by DIGEST-MD5 method.
- Changed DNs in exmple to be dc=example,dc=com.

Section 10

- Updated consideration on use of cleartext passwords to include other unprotected authentication credentials

- Substantial rework of consideration on misuse of unauthenticated bind.

F.9. Changes for draft-ldapbis-authmeth-10

- Reorganized content of sections 3-9 to improve document flow and reduce redundancy.
- Resolved issue of effect of Start TLS and TLS closure on association state.
- Made numerous minor wording changes based on WG feedback.
- Updated list of threats for Section 1.
- Recommendation that servers should not support weaker TLS ciphersuites unless other protection is in place.
- Moved authentication state table to appendix and relettered appendices.

F.10. Changes for draft-ldapbis-authmeth-11

General

- Many editorial changes throughout to clarify wording and better express intent, primarily based on suggestions from WG mail list.
- More standard naming of authentication mechanisms throughout document, e.g. "Anonymous Authentication Mechanism of the Simple Bind Choice".

Section 1

- Editorial changes to add clarity.
- Moved section 2 of authmeth -09 into section 1

Section 2

- New section outlining implementation requirements.

Section 3.1.1

- Editorial clarification on need for following operation sequencing requirements.

Section 3.1.4

- New section added to describe use of client certificates with StartTLS. Incorporates material moved from other sections of authmeth -09.

Section 4

- New section added to discuss associations. Related material was moved from various other sections of authmeth -09 and incorporated into this new section.

Section 5

- Added several paragraphs regarding transmission and derivation of authentication and authorization identities using the Bind

operation.

Section 8

- Clarified rules for determining valid credentials and situations where invalidCredentials result is to be returned.

Section 14

- Added three security considerations based on WG feedback.

Appendix A

- Simplfied state tables by removing two unnecessary actions from the actions table, and removing the current state column of the

state transition table. Updated references to authmeth and [Protocol].

F.11. Changes for draft-ldapbis-authmeth-12

General

- Changed refererences from Start TLS to StartTLS.
- Removed Appendix B: Example Deployment Scenarios
- Removed Appendix H as all issues listed in the appendix are now resolved.

Section 2

- Added implementation requirement that server implementations that SUPPORT StartTLS MUST support the TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA ciphersuite.

Section 3.1.2

- Added wording clarifying that a client's association is unaffected if a non-success resultCode is returned in the StartTLS response.

Section 9.2

- Final paragraph of this section details requirements for serverSaslCreds field when no challenge value is sent.

Section 10

- Clarified language on uAuthzID usage.

Section 12

- Moved entire section into security considerations. New section number is 12.1.1.

- Reorganized security considerations by topic.
- Added several security considerations based on WG feedback.

Section 13

- Moved section to become section 3.3.

F.12. Changes for draft-ldapbis-authmeth-13

General

- General edits for clarity and to remove errors.
- Reworded definition of association (section 1.2) and reworked usage of association throughout document. Current semantics: every connection has an association with the same lifetime as the connection, and that association passes through various authorization states.

- Made usage of data confidentiality consistent throughout document.

Section 1
- Reworded mechanisms 3 and 4 for more parallelism.
- Changed language on rationale for required mechansisms from future to past tense.

Section 2
- Clarified that implementations may support any additional authentication mechanism, not just mechanisms associated with simple and SASL bind choices.

Section 3
- Moved paragraph explaining goals for using TLS with LDAP from security considerations to here.

Section 4.3
- Reworked text to better explain meaning of strongAuthRequired result code when for invalidated associations.

Section 8
- Clarified action when simple bind request has a DN with invalid syntax.

Section 12.1
- Added ability to configure and enforce administrative service limits as a way to protect against denial of service attacks.

Section 12.2
- Clarified that this security consideration relates to performing client authentication during the TLS handshake and not to subsequent SASL EXTERNAL authentication.

Appendix A
- Updated tables by collapsing identical states and actions. Also added an invalidated association state and accompanying actions.

Added implementation requirement that server implementations

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.

Full Copyright Statement