Network Working Group M. Wahl
INTERNET-DRAFT Critical Angle Inc.
Expires in six months from 24 March 1997

X.500 Strong Authentication Mechanisms for LDAPv3
<draft-ietf-asid-ldapv3-strong-00.txt>

1. Status of this Memo

2. Abstract

This document defines two SASL [1] authentication mechanisms which may be used with LDAPv3 [2]. These mechanisms are only for authentication, they have no effect on the protocol encodings and are not designed to provide integrity or confidentiality services.

3. Model

Two mechanisms are defined, which are equivalent to the "protected" password and "strong" mechanisms of X.500. Unprotected password authentication is done using the existing LDAP "simple" bind, not with SASL. These mechanisms may also be used in other SASL-based protocols.

The client may include one of these mechanisms and its credential in the BindRequest.

The server will return a BindResponse with one of the following result codes:

- success, and the serverCreds field absent, implying that the server successfully authenticated the client but is not returning any authentication information about the server;

- success, and the serverCreds field present, with the same mechanism as that requested by the client, and the credentials of the server itself;

- protocolError, if the server does not implement LDAP version 3,

- authMethodNotSupported, if the server does not implement this mechanism;

- strongAuthRequired, referral, inappropriateAuthentication, invalidCredentials, busy or unavailable, if the server did not successfully authenticate the client.

If the server supports either of these mechanisms, the mechanism name(s) must be included as values in the root DSE attribute supportedSASLMechanisms.

4. Encoding Requirements

This document describes data elements using ASN.1 structures, which are encoding using a subset of the Basic Encoding Rules, as done in LDAPv3 [2]. Implementations must follow the encoding restrictions of LDAP, and additional encoding restrictions apply to the elements defined in this specification:

- BIT STRING values are to be encoded in primitive form only. Unused bits in the final octet of the encoding of a BIT STRING value, if there are any, should always be set to zero.

- UTC Times must be encoded with the "Z" suffix, not as a local time.

5. X.511-Protected

The "X.511-Protected" authentication mechanism allows a hash of the password, combined optionally with the current time and random numbers, to be sent to or returned from the server. The protected field contains the hash value. This prevents a password from being carried in the clear.

The mechanism field is set to the string "X.511-Protected", and the credentials field contain the DER encoding of a value of the following ASN.1 type:

```
ProtectedPassword ::= SEQUENCE {
        time1                   [0] UTCTime OPTIONAL,
        time2                   [1] UTCTime OPTIONAL,
        random1                 [2] BIT STRING OPTIONAL,
        random2                 [3] BIT STRING OPTIONAL,
        algorithmIdentifier     [4] LDAPOID,
        encipheredPassword      [5] BIT STRING }
```

The use of the time1, time2, random1, random2 and encipheredPassword fields are as defined in ITU-T Rec. X.509 [3] and the functional profile for X.500 for the environment in which this authentication mechanism is to be used.

The algorithmIdentifier must be an entirely numeric string representation of an OBJECT IDENTIFIER.

The name field of the BindRequest must be a nonempty string when this mechanism is being used to authenticate the client. Note that this security mechanism is not intended to protect against attackers modifying the bind name field or other protocol elements.

6. X.511-Strong

Strong authentication to the directory can be accomplished using the "X.511-Strong".

The mechanism field is set to the string "X.511-Strong", and the credentials field set to a DER-encoding of a value of the following ASN.1 type:

```
     StrongCredentials ::= SEQUENCE {
             certification-path      [0] AF.CertificationPath OPTIONAL,
             bind-token              [1] DAS.Token }
```

The ASN.1 type "CertificationPath" is defined in X.509 [3], and the ASN.1 type "Token" is defined in X.511 [4]. The procedures for generation and validation of the bind token are defined in X.509 and X.511.

When the credentials are being used to authenticate the client, it is recommended that the certification-path field be present, which will contain minimally the client's certificate. If the certification-path field is supplied, then the name field of the BindRequest must be an empty string, and the server will obtain the name of the client from the subject field of the certification-path userCertificate.

It is recommended for interoperability that if the server's or client's certificates contain RSA public keys, the PKCS md5WithRSAEncryption (1.2.840.113549.1.1.4) algorithm should be used.

7. Attributes in the Root DSE

This document defines three attributes which may be present in the server's root DSE.

7.1. Checking the Current Time

With these mechanisms, authentication between the client and the server may fail due to a lack of clock synchronization. This may be detected by the client, by reading the currentTime attribute.

This attribute has a single value, a string containing a GeneralizedTime character string. This attribute need only be present if the server supports LDAP strong or protected simple authentication. Otherwise if the server does not know the current time, or does not choose to present it to clients, this attribute need not be present. The client may wish to use this value to detect whether a strong or protected bind is failing because the client and server clocks are not sufficiently synchronized, but clients must not use this time field for setting their own system clock.

The definition of the attribute is:

```
 ( 1.3.6.1.4.1.1466.101.120.2 NAME 'currentTime'
   SYNTAX 'GeneralizedTime' SINGLE-VALUE USAGE dSAOperation )
```

7.2. Validating the Name of the Server

A server which accepts binds with the X.511-Strong mechanism must have a Distinguished Name, which preferably should uniquely identify it.

A client may check that the Distinguished Name which it has for a server matches that which the server is expecting by reading the serverName;binary attribute from the servers' root DSE.

This attribute's value is the server's Distinguished Name. The attribute will likely be absent if the server does not accept strong authentication using X.511-Strong. However, the presence of this attribute does not guarantee that the server will be able to perform strong authentication.

If the server acts as a gateway to gateway to more than one X.500 DSA capable of strong authentication, there may be multiple values of this attribute, one per DSA.

(Note: this attribute is distinct from myAccessPoint, for it is not required that a server have a presentation address in order to perform strong authentication.)

It is likely that clients will retrieve this attribute in binary. If all attributes of the root DSE are requested, servers must return the attribute values in binary. The binary value is the octets of a DER-encoded value of an X.501 DistinguishedName type, e.g. the first byte is a SEQUENCE tag, and so on.

Client implementors should be aware that values returned by the server may be modified in transit.

The definition of this attribute is:

```
( 1.3.6.1.4.1.1466.101.120.3 NAME 'serverName'
 SYNTAX 'DN' USAGE dSAOperation )
```

7.3. Obtaining the Certification Path of the Server

A server which accepts binds with the X.511-Strong mechanism may have certification paths, and this information may be of use to the client in determining a common point of trust.

A client may retrieve a server's certification paths by reading the certificationPath;binary attribute from the server's root DSE.

An attribute value contains a binary DER encoding data type, which is the certificate path for a server. If the server does not have a certificate path this attribute must be absent.

Clients must only retrieve this attribute in binary. If all attributes of the root DSE are requested, servers must return the attribute values in binary. The binary value is the octets of a DER-encoded value of an X.509 CertificationPath type, e.g. the first byte is a SEQUENCE tag, and so on.

The definition of this attribute is:

```
( 1.3.6.1.4.1.1466.101.120.4 NAME 'certificationPath'
 SYNTAX 'CertificatePath' USAGE dSAOperation )
```

7.4. Determining Supported Algorithms

The server may list the names of algorithms it supports for use in these mechanisms in the supportedAlgorithms attribute of the root DSE.

```
( 2.5.4.52 NAME 'supportedAlgorithms' SYNTAX 'SupportedAlgorithm' )
```

8. Security Considerations

These algorithms are designed to be used for authentication where the underlying transport service cannot guarantee confidentiality.

It should be noted that the name field of the BindRequest is not protected against modification in the "X.511-Protected" mechanism.

These mechanisms do not provide for confidentiality of any data transferred between the client and the server, except for the password in the "X.511-Protected" mechanism. These mechanisms do not prevent an authenticated association from being hijacked.

8. Acknowledgements

Design ideas included in this document are based on those from ITU and ISO, and the IETF ASID Working Groups. The contributions of individuals in these working groups is gratefully acknowledged.

9. Bibliography

[1] J. Meyers, "Simple Authentication and Security Layer", INTERNET-DRAFT <draft-myers-auth-sasl-04.txt>, July 1996.

[2] M. Wahl, T. Howes, S. Kille, "Lightweight Directory Access Protocol (v3)", INTERNET-DRAFT <draft-ietf-asid-ldapv3-protocol-04.txt>, February 1997.

[3] ITU-T Rec. X.509, "The Directory: Authentication Framework", 1993.

[4] ITU-T Rec. X.511, "The Directory: Abstract Service Definition", 1993.

10. Authors' Address

Mark Wahl
Critical Angle Inc.
4815 W Braker Lane #502-385
Austin, TX 78759
USA

EMail: M.Wahl@critical-angle.com