LDAP-EXT Working Group
INTERNET-DRAFT
Expires in six months
Intended Category: Informational

Valerie Chu
Netscape Communications Corp.

December 1998

Password Policy for LDAP Directories
<draft-vchu-ldap-pwd-policy-00.txt>

1. Status of this Memo

This document is an Internet-Draft. 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.''

To view the entire list of current Internet-Drafts, please check the "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), ftp.nordu.net (Northern Europe), ftp.nic.it (Southern Europe), munnari.oz.au (Pacific Rim), ftp.ietf.org (US East Coast), or ftp.isi.edu (US West Coast).

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.

2. Abstract

This document describes the implementation of password policy in Netscape LDAP directories, and introduces two new object classes, twenty-three new attribute types, and two new controls in support of password policy.

Password policy is a set of rules that control how passwords are used in LDAP directories. In order to improve the security of LDAP directories and make it difficult for password cracking programs to break into directories, it is desirable to enforce a set of rules on password usage. These rules are made to ensure that the users change their passwords periodically, the new password meets construction requirements, the re-use of the old password is restricted, and lock out the users

after a certain number of bad password attempts.

3. Overview

LDAP-based directory services currently are accepted by many organizations as the access protocol for directories. The ability to ensure the secure read, update access to directory information throughout the network is essential to the successful deployment. There are several security mechanisms which are used in Netscape LDAP implementation to protect the directory data. For example, the access control is used to prevent unauthorized access to information stored in directories; SASL is used to negotiate for integrity and privacy services.[RFC-2251] The most fundamental security mechanism in Netscape Directory is the simple authentication using password. In many systems, in order to improve the security of the system, the simple password-based authentication often is used in conjunction with a set of password restrictions to control how passwords are used in the system. For example, the passwd program in UNIX systems, or the user account policy in WindowsNT, has a set of rules that users need to follow to use password authentication. At the moment, LDAP does not define a password policy model, but it is needed to achieve greater security protection and it is critical to the successful deployment of LDAP directories.

Specifically, the password policy defines:

- The maximum length of time that a given password is valid.

- The minimum length of time required between password changes.

- The maximum length of time before a user's password is due to expire that the user will be sent a warning message.

- Whether users can reuse passwords.

- The minimum number of characters a password must contain.

- Whether the password syntax is checked before a new password is saved.

- Whether users are allowed to change their own passwords.

- Whether passwords must be changed after they are reset by the administrator.

- Whether users will be locked out of the directory after a given number of failed bind attempts.

- How long users will be locked out of the directory after a given number of failed bind attempts.

- The length of time before the password failure counter which keeps track of the number of failed password attempts is reset.

The password policy defined in this document is applied to the LDAP simple authentication method [RFC-2251] and userPassword attribute values only.

In this document, the term "user" represents any application which is an LDAP client using the directory to retrieve or store information.

Directory administrators are not forced to comply with any of password policies.

4. New Attribute Types and Object Classes

4.1. The passwordPolicy Object Class

The passwordPolicy object class holds the password policy settings for a set of user accounts. In the Netscape Directory implementation, they are located in the "cn=config" entry.

The description of passwordPolicy object class:

```
   ( 2.16.840.1.113730.3.2.13
     NAME 'passwordPolicy'
     AUXILIARY
     SUP top
     DESC 'Password Policy object class to hold password policy information'
     MAY (
           passwordMaxAge $ passwordExp $ passwordMinLength $
           passwordKeepHistory $ passwordInHistory $ passwordChange $
           passwordCheckSyntax $ passwordWarning $ passwordLockout $
           passwordMaxFailure $ passwordUnlock $ passwordLockoutDuration $
           passwordMustChange $ passwordStorageScheme $ passwordMinAge $
           passwordResetFailureCount
         )
   )
```

4.2. The new attribute types used in the passwordPolicy Object Class:

```
   ( 2.16.840.1.113730.3.1.97
     NAME 'passwordMaxAge'
     DESC 'the number of seconds after which user passwords will expire'
     EQUALITY 'caseIgnoreMatch'
     SYNTAX '1.3.6.1.4.1.1466.115.121.1.15'
```

```
   )
   ( 2.16.840.1.113730.3.1.98
     NAME 'passwordExp'
     DESC 'a flag which indicates whether passwords will expire after a
           given number of seconds'
     EQUALITY 'caseIgnoreMatch'
     SYNTAX '1.3.6.1.4.1.1466.115.121.1.15'
   )
   ( 2.16.840.1.113730.3.1.99
     NAME 'passwordMinLength'
     DESC 'the minimum number of characters that must be used in a password'
     EQUALITY 'caseIgnoreMatch'
     SYNTAX '1.3.6.1.4.1.1466.115.121.1.15'
   )
   ( 2.16.840.1.113730.3.1.100
     NAME 'passwordKeepHistory'
     DESC 'a flag which indicates whether passwords can be reused"
     EQUALITY 'caseIgnoreMatch'
     SYNTAX '1.3.6.1.4.1.1466.115.121.1.15'
   )
   ( 2.16.840.1.113730.3.1.101
     NAME 'passwordInHistory'
     DESC 'the number of passwords the directory server stores in history'
     EQUALITY 'caseIgnoreMatch'
     SYNTAX '1.3.6.1.4.1.1466.115.121.1.15'
   )
   ( 2.16.840.1.113730.3.1.102
     NAME 'passwordChange'
     DESC 'a flag which indicates whether users can change their passwords'
```

```
      EQUALITY 'caseIgnoreMatch'
      SYNTAX '1.3.6.1.4.1.1466.115.121.1.15'
    )
    ( 2.16.840.1.113730.3.1.103
      NAME 'passwordCheckSyntax'
      DESC 'a flag which indicates whether the password syntax will be checked
            before the password is saved'
      EQUALITY 'caseIgnoreMatch'
      SYNTAX '1.3.6.1.4.1.1466.115.121.1.15'
    )
    ( 2.16.840.1.113730.3.1.104
      NAME 'passwordWarning'
      DESC 'the number of seconds before a user's password is due to expire that
            the user will be sent a warning message'
      EQUALITY 'caseIgnoreMatch'
      SYNTAX '1.3.6.1.4.1.1466.115.121.1.15'
    )
    ( 2.16.840.1.113730.3.1.105
      NAME 'passwordLockout'
```

```
      DESC 'a flag which indicates whether users will be locked out of the
            directory after a given number of consecutive failed bind attempts'
      EQUALITY 'caseIgnoreMatch'
      SYNTAX '1.3.6.1.4.1.1466.115.121.1.15'
    )
    ( 2.16.840.1.113730.3.1.106
      NAME 'passwordMaxFailure'
      DESC 'the number of consecutive failed bind attempts after which a user
            will be locked out of the directory'
      EQUALITY 'caseIgnoreMatch'
      SYNTAX '1.3.6.1.4.1.1466.115.121.1.15'
    )
    ( 2.16.840.1.113730.3.1.108
      NAME 'passwordUnlock'
      DESC 'a flag which indicates whether a user will be locked out of the
            directory for a given number of seconds or until the administrator
            resets the password after an account lockout'
      EQUALITY 'caseIgnoreMatch'
      SYNTAX '1.3.6.1.4.1.1466.115.121.1.15'
    )
    ( 2.16.840.1.113730.3.1.109
      NAME 'passwordLockoutDuration'
```

```
     DESC 'the number of seconds that users will be locked out of the directory
           after an account lockout
     EQUALITY 'caseIgnoreMatch'
     SYNTAX '1.3.6.1.4.1.1466.115.121.1.15'
   )
   ( 2.16.840.1.113730.3.1.220
     NAME 'passwordMustChange'
     DESC 'a flag which indicates whether users must change their passwords whe
           they first bind to the directory server'
     EQUALITY 'caseIgnoreMatch'
     SYNTAX '1.3.6.1.4.1.1466.115.121.1.15'
   )
   ( 2.16.840.1.113730.3.1.221
     NAME 'passwordStorageScheme'
     DESC 'the type of hash algorithm used to store directory server passwords'
     EQUALITY 'caseIgnoreMatch'
     SYNTAX '1.3.6.1.4.1.1466.115.121.1.15'
   )
```

The description of password storage scheme can be found in [RFC-2307].

```
   ( 2.16.840.1.113730.3.1.222
     NAME 'passwordMinAge'
     DESC 'the number of seconds that must elapse before a user can change thei
           password again'
     EQUALITY 'caseIgnoreMatch'
     SYNTAX '1.3.6.1.4.1.1466.115.121.1.15'
   )
```

```
   ( 2.16.840.1.113730.3.1.223
     NAME 'passwordResetFailureCount'
     DESC 'the number of seconds after which the password failure counter will
           be reset'
     EQUALITY 'caseIgnoreMatch'
     SYNTAX '1.3.6.1.4.1.1466.115.121.1.15'
   )
```

Currently in Netscape Directory password policy implementation, passwordMaxAge, passwordMinLength, passwordInHistory, passwordWarning, passwordMaxFailure, passwordLockoutDuration, passwordMinAge, and passwordResetFailureCount attributes are defined as 1.3.6.1.4.1.1466.115.121.1.15 ('Directory String'). It is recommended to change them to 1.3.6.1.4.1.1466.115.121.1.27 ('Integer') in the future implementation.

The attributes which are used as a flag have the syntax '1.3.6.1.4.1.1466.115.121.1.15' ('Directory String'). A value of '1' represents 'true', while '0' represents 'false'. It is recommented to change them to 1.3.6.1.4.1.1466.115.121.1.7 ('Boolean') in the future implementation.

4.3. The passwordObject Object Class

The passwordObject object class holds the password policy state information for each user. For example, how many consecutive bad password attempts an user made. The information is located in each user entries. The description of passwordObject object class:

```
   ( 2.16.840.1.113730.3.2.12
     NAME 'passwordObject'
     AUXILIARY
     SUP top
     DESC 'Password object class to hold password policy information for each
           entry'
     MAY (
           passwordExpirationTime $ passwordExpWarned $ passwordRetryCount $
           retryCountResetTime $ accountUnlockTime $ passwordHistory $
           passwordAllowChangeTime
         )
   )
```

4.4. The new attribute types used in the passwordObject Object Class:

```
   ( 2.16.840.1.113730.3.1.91
     NAME 'passwordExpirationTime'
     DESC 'the time the entry's password expires'
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
     EQUALITY generalizedTimeMatch
```

```
     ORDERING generalizedTimeOrderingMatch
     SINGLE-VALUE
     USAGE directoryOperation
   )
   ( 2.16.840.1.113730.3.1.92
     NAME 'passwordExpWarned'
     DESC 'a flag which indicates whether a password expiration warning is sent
           to the client'
     EQUALITY 'caseIgnoreMatch'
     SYNTAX '1.3.6.1.4.1.1466.115.121.1.15'
     SINGLE-VALUE
     USAGE directoryOperation
   )
   ( 2.16.840.1.113730.3.1.93
     NAME 'passwordRetryCount'
     DESC 'the count of consecutive failed password attempts'
     EQUALITY 'caseIgnoreMatch'
     SYNTAX '1.3.6.1.4.1.1466.115.121.1.15'
     SINGLE-VALUE
     USAGE directoryOperation
   )
   ( 2.16.840.1.113730.3.1.94
     NAME 'retryCountResetTime'
     DESC 'the time to reset the passwordRetryCount'
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
     EQUALITY generalizedTimeMatch
     ORDERING generalizedTimeOrderingMatch
     SINGLE-VALUE
     USAGE directoryOperation
   )
   ( 2.16.840.1.113730.3.1.95
     NAME 'accountUnlockTime'
     DESC 'the time that the user can bind again after an account lockout'
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
     EQUALITY generalizedTimeMatch
     ORDERING generalizedTimeOrderingMatch
     SINGLE-VALUE
     USAGE directoryOperation
   )
   ( 2.16.840.1.113730.3.1.96
     NAME 'passwordHistory'
     DESC 'the history of user's passwords'
     SYNTAX 1.3.6.1.4.1.1466.115.121.1.5
     EQUALITY bitStringMatch
     USAGE directoryOperation
   )
   ( 2.16.840.1.113730.3.1.214
     NAME 'passwordAllowChangeTime'
```

```
     DESC 'the time that the user is allowed change the password'
```

```
      SYNTAX 1.3.6.1.4.1.1466.115.121.1.24
      EQUALITY generalizedTimeMatch
      ORDERING generalizedTimeOrderingMatch
      SINGLE-VALUE
      USAGE directoryOperation
    )
```

5. Password Expiration and Expiration Warning

New attributes, passwordExp, passwordMaxAge, and passwordWarning are defined to specify whether the password will expire, when the password expires and when a warning message will be sent to the client respectively. The actual expiration time for a password will be stored in a new attribute, passwordExpirationTime attribute in the user entry.

After bind operation succeed with authentication, the server should check for password expiration. If the password expiration policy is on and the account's password is expired, the server should send bindResponse with the resultCode: LDAP_INVALID_CREDENTIALS along with an error message to inform the client that the password has expired. If the password is going to expire sooner than the password warning duration, the server should send bindResponse with the resultCode: LDAP_SUCCESS, and should include the password expiring control in the controls field of the bindResponse message:

```
controlType:  2.16.840.1.113730.3.4.5,

controlValue: an octet string to indicate the time in seconds until
              the password expires.

criticality:  false
```

The server should send at least one warning message to the client before expiring the client's password.

6. Password Minimum Age

This policy defines the number of seconds that must pass before a user can change the password again. This policy can be used in conjunction with the password history policy to prevent users from quickly cycling through passwords in history so that they can reuse the old password. A value of zero indicates that the user can change the password immediately.

During the modify password operation, the server should check if the user is allowed to change password at this time. If not, the server should send the LDAP_CONSTRAINT_VIOLATION result code back to the client and an error message to indicate that the password cannot be changed within password minimum age.

7. Password History

passwordHistory and passwordInHistory attributes control whether the user can reuse passwords and how many passwords the directory server stores in history.

During the modify password operation, the server should check for password history. If password history is on and the new password matches one of the old passwords in history, the server should send modifyResponse back to the client with resultCode: LDAP_CONSTRAINT_VIOLATION, and an error message to indicate the new password is in history, choose another password.

8. Password Syntax and Minimum length

The passwordCheckSyntax attribute indicates whether the password syntax will be checked before a new password is saved. If this policy is on, the directory server should check that the new password meets the password minimum length requirement and that the string does not contain any trivial words such as the user's name, user id and so on.

The passwordMinLength attribute defines the minimum number of characters that must be used in a password.

During the modify or add password operation, the server should check for password syntax. If password check syntax is on and the new password fail the syntax checking, the server should send modifyResponse or addResponse back to the client with resultCode: LDAP_CONSTRAINT_VIOLATION, and an error message to indicate the new password failed the syntax checking, the user should choose another password.

9. User Defined Passwords

This policy defines whether the users can change their own passwords. During the modify password operation, the server should check if the user is allowed to change password. If not, the server should send to the client the LDAP_UNWILLING_TO_PERFORM result code and an error message to indicate that the user is not allowed to change password.

10. Password Change After Reset

This policy forces the user to select a new password on first bind or after password reset. After bind operation succeed with authentication,

the server should check if the password change after reset policy is on and this is the first time logon. If so, the server should send bindResponse with the resultCode: LDAP_SUCCESS, and should include the password expired control in the controls field of the bindResponse message:

```
    controlType:  2.16.840.1.113730.3.4.4,

    controlValue: an octet string: "0",

    criticality: false
```

After that, for any operation issued by the user other than modify password, bind, unbind, abandon, or search, the server should send the response message with the resultCode: LDAP_UNWILLING_TO_PERFORM, and should include the password expired control in the controls field of the response message:

```
    controlType:  2.16.840.1.113730.3.4.4,

    controlValue: an octet string: "0",

    criticality: false
```

11. Password Guessing limit

This policy enforces the limit of number of tries the client has to get the password right. The user will be locked out of the directory after a given number of consecutive failed attempts to bind to the directory. This policy protects the directory from automated guessing attacks.

The server should keep a failure counter in the passwordRetryCount attribute for each entry. The server should increment the failure counter when a bind operation fails with the LDAP_INVALID_CREDENTIALS error code. The server should clear the failure counter when a bind operation succeeds with authentication, the account password is reset by administrator, or when the failure counter reset time is reached.

During the bind operation, the server should check for password guessing limit. If password guessing limit policy is on and the password guessing limit is reached, the server should send bindResponse back to the client with resultCode: LDAP_CONSTRAINT_VIOLATION, and an error message to indicate the password failure limit is reached.

12. Server Implementation

12.1. Password policy initialization

The passwordPolicy object class holds the password policy settings for a set of user accounts. During the server initial startup, password policy should be assigned a set of initial values. The settings should be modified only by the directory administrators and should be readable by anyone. The server should preserve the settings over server restart. Currently in the Netscape Directory implementation, the password policy settings are stored in "cn=config" entry and an identical copy is kept in a configuration file which is used as bootstrap. The Netscape Directory password default settings are listed below as an example.

- User may change password

- Do not need to change password first time logon

- Use SHA as the password hash algorithm

- No password syntax check

- Password minimum length: 6

- No password expiration

- Expires in 100 days

- No password minimum age

- Send warning one day before password expires

- Do not keep password history

- Six passwords in history

- No account lockout

- Lockout after 3 bind failures

- Do not lockout forever

- Lock account for 60 minutes

- Reset retry count after 10 minutes

In ldif format:

```
passwordchange: on
```

```
passwordmustchange: off

passwordstoragescheme: SHA

passwordchecksyntax: off

passwordminlength: 6

passwordexp: off

passwordmaxage: 8640000

passwordminage: 0

passwordwarning: 86400

passwordkeephistory: off

passwordinhistory: 6

passwordlockout: off

passwordmaxfailure: 3

passwordunlock: on
```

passwordlockoutduration: 3600

passwordresetfailurecount: 600

12.2. Bind Operations

12.2.1. During bind operations, the server should check for password guessing limit. If password guessing limit policy is on and the password guessing limit is reached, the server should send bindResponse back to the client with resultCode: LDAP_CONSTRAINT_VIOLATION, and an error message to indicate the password failure limit is reached. Otherwise the server should continue the bind operation.

12.2.2. After Bind Operations succeed with authentication, the server should

1. Clear the password failure counter.

2. Check if the password change after reset policy is on and this is the first time logon. If so, the server should disallow all operations issued by this user except modify password, bind , unbind, abandon, or search. The server should send bindResponse

with the resultCode: LDAP_SUCCESS, and should include the password expired control in the controls field of the bindResponse message.

controlType: 2.16.840.1.113730.3.4.4,

controlValue: an octet string: "0",

criticality: false

3. Check for password expiration. If the password expiration policy is on and the account's password is expired, the server should send bindResponse with the resultCode: LDAP_INVALID_CREDENTIALS along with an error message to inform the client that the password has expired.

4. Check if the password is going to expire sooner than the password warning duration, the server should send bindResponse with the

resultCode: LDAP_SUCCESS, and should include the password expiring control in the controls field of the bindResponse message:

```
controlType:  2.16.840.1.113730.3.4.5,

controlValue: an octet string to indicate the time in seconds
              until the password expires.

criticality:  false
```

12.2.3. After Bind Operations fail with LDAP_INVALID_CREDENTIALS, the server should

1. Check if it is time to reset the password failure counter. If so, set the failure counter to 1 and re-calculate the next failure counter reset time. Otherwise, increment the failure counter.

2. Check if failure counter exceeds the allowed maximum value. If so, the server should lock the user account.

12.3. Add Password Operations

12.3.1. During the add password operation, the server should

1. Check for password syntax. If password check syntax is on and the new password fail the syntax checking, the server should send addResponse back to the client with resultCode: LDAP_CONSTRAINT_VIOLATION, and an error message to indicate the

new password failed the syntax checking, the user should choose another password.

2. Calculate and add passwordexpirationtime and passwordallowchangetime attributes to the entry if password expiration policy and password minimum age policy are on respectively.

12.4. Modify Password Operations

12.4.1. During the modify password operation, the server should

1. Check if the user is allowed to change password. If not, the server should send to the client the LDAP_UNWILLING_TO_PERFORM result code and an error message to indicate that the user is not allowed to change password.

2. Check for password minimum age, password minimum length, password history, and password syntax. If the checking fails, the server should send modifyResponse back to the client with resultCode: LDAP_CONSTRAINT_VIOLATION, and an appropriate error message.

3. If it is the first time logon and the user needs to change password the first time logon, the server should check if the userpassword attribute is in this modify request. If so, the server should continue the modify operation. Otherwise, the server should send the response message with the resultCode: LDAP_UNWILLING_TO_PERFORM, and should include the password expired control in the controls field of the response message:

   controlType: 2.16.840.1.113730.3.4.4,

   controlValue: an octet string: "0",

   criticality: false

12.4.2. After modify password operations succeed, the server should

1. Update password history in the user's entry, if the password history policy is on.

2. Update passwordExpirationTime in the user's entry, if the password expiration policy is on.

3. Update passwordAllowChangeTime in the user's entry, if the password minimum age policy is on.

4. Clear the password failure counter, if the password is reset by a directory administrator.

5. Set a flag to indicate the user is the first time logon, if the password change after reset policy is on and the password is reset by a directory administrator.

13. Client Implementation

13.1. Bind Response

For every bind response received, the client needs to parse the bind result code, error message, and controls to determine if any of the following conditions is true and prompt the user accordingly.

1. The user needs to change password first time logon. The user should be prompted to change the password immediately.

```
resultCode: LDAP_SUCCESS, with the control
    controlType: 2.16.840.1.113730.3.4.4,
    controlValue: "0",
    criticality: false
```

2. This is a warning message that the server sends to a user to indicate the time in seconds until the user's password expires.

```
resultCode: LDAP_SUCCESS, with the control
    controlType:  2.16.840.1.113730.3.4.5,
    controlValue: an octet string to indicate the time in seconds until
                  the password expires.
    criticality:  false
```

3. The password failure limit is reached. The user needs to retry later or contact the directory administrator to reset the password.

```
resultCode: LDAP_CONSTRAINT_VIOLATION, with an appropriate error message.
           For example:
           errorMessage: "exceed password retry limit"
```

4. The password is expired. The user needs to contact the directory administrator to reset the password.

```
resultCode: LDAP_INVALID_CREDENTIALS, with an appropriate error message.
           For example:
           errorMessage: "password expired"
```

13.2. Modify Responses

For the modify response received for the change password request, the client needs to check the result code and error message to determine if it failed the password checking, and either let the user retry or quit.

1. The user defined password policy is disabled. The user is not allowed to change password.

```
    resultCode: LDAP_UNWILLING_TO_PERFORM, with an appropriate error message.
                For example:
                errorMessage: "user is not allowed to change password"
```

2. The new password failed the password syntax checking, or the current password has not reached the minimum password age, or the new password is in history.

```
    resultCode: LDAP_CONSTRAINT_VIOLATION, with an appropriate error message.
                For example:
                errorMessage: "invalid password syntax"
                errorMessage: "password in history"
                errorMessage: "trivial password"
                errorMessage: "within minimum password age"
```

13.3. Add Responses

For the add response received for the add entry request, the client needs to check the result code and error message to determine if it failed the password checking, and either let the user retry or quit.

1. The new password failed the password syntax checking.

```
    resultCode: LDAP_CONSTRAINT_VIOLATION, with an appropriate error message.
                For example:
                errorMessage: "invalid password syntax"
                errorMessage: "trivial password"
```

13.4. Other Responses

For operations other than bind, unbind, abandon, or search, the client needs to check the following result code and control to determine if the user needs to change the password immediately.

1. The user needs to change password first time logon. The user should be prompted to change the password immediately.

```
     resultCode: LDAP_UNWILLING_TO_PERFORM, with the control
```

```
         controlType: 2.16.840.1.113730.3.4.4,
         controlValue: "0",
         criticality: false
```

14. Security Considerations

The password policy defined in this document is applied to the LDAP simple authentication method [RFC-2251] and userPassword attribute values only. The simple authentication method provides minimal authentication facilities, with the contents of the authentication field consisting only of a cleartext password. Note that the simple authentication method and password policy are designed for authentication where the underlying transport service cannot guarantee confidentiality. Use of simple authentication method and password policy may result in disclosure of the password to unauthorized parties. SASL and TLS mechanisms may be used with LDAP to provide integrity or confidentiality services.

15. Bibliography

[RFC-2251]Wahl, M., Howes, T., Kille, S., "Lightweight Directory Access Protocol (v3)", RFC 2251, August 1997.

[RFC-2307]L. Howard, "An Approach for Using LDAP as a Network Information Service", RFC 2307, March 1998.

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

16. Author's Addresses

Valerie Chu
Netscape Communications Corp.
501 E. Middlefield Rd.
Mountain View, CA 94043
USA
+1 650 937-3443
vchu@netscape.com