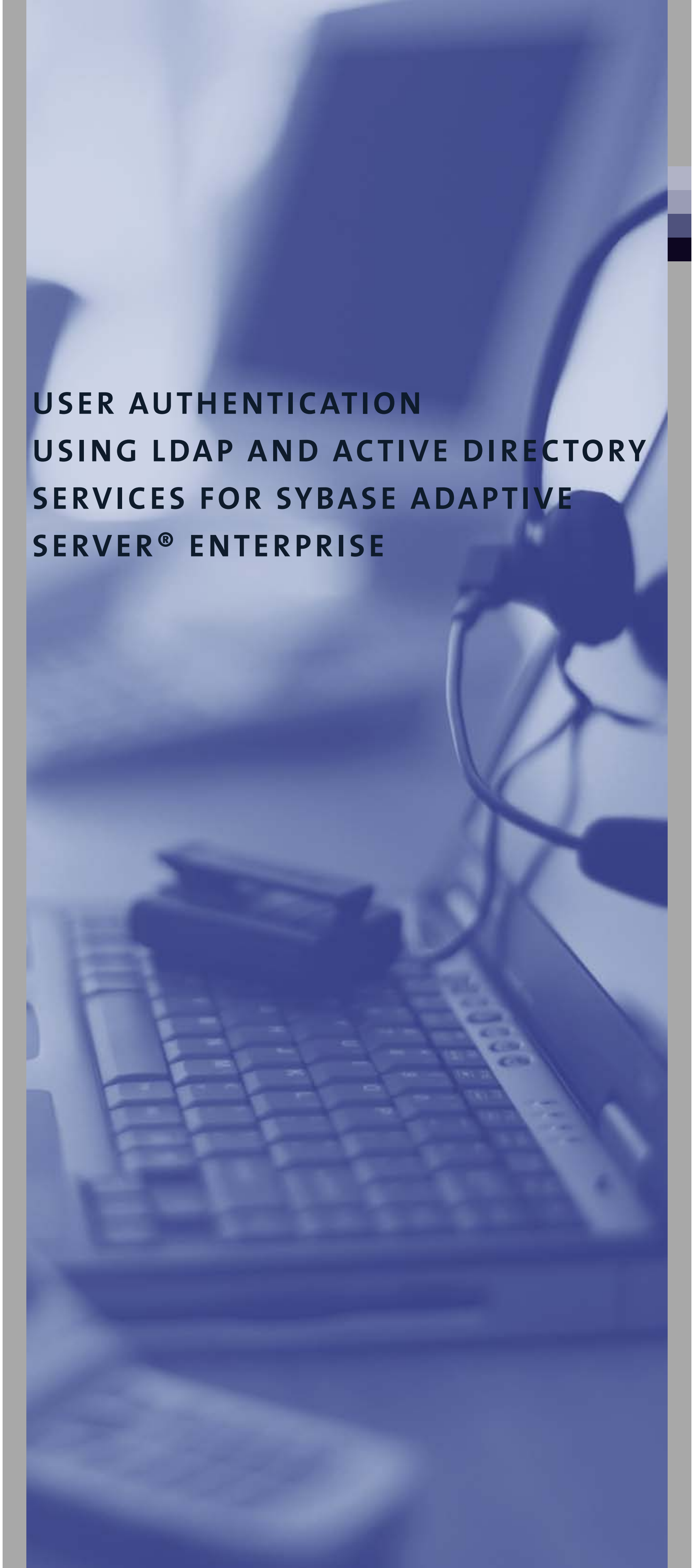
SYBASE®



USER AUTHENTICATION USING LDAP AND ACTIVE DIRECTORY SERVICES FOR SYBASE ADAPTIVE SERVER® ENTERPRISE

TECHNICAL WHITE PAPER

TABLE OF CONTENTS

LDAP USER AUTHENTICATION

SUMMARY

Lightweight Directory Access Protocol (LDAP) is an industry standard for accessing directory services over a network. Adaptive Server Enterprise (ASE) version 12.5.1, introduced the ability to use an LDAP Server for user authentication. Adaptive Server Enterprise version 12.5.2 further extends user authentication to Active Directory and improves authentication controls.

The primary benefits of using LDAP to manage users are:

- **centralized password security policies in one authority,**
- **centralized identity and passwords across both UNIX and Windows,**
- **simplified creation and deletion of users,**
- **simplified user password for both the operating system and application, and**
- **reduced overall cost of ownership.**

LDAP USER AUTHENTICATION – THE BUSINESS CASE

SIMPLIFIED ADMINISTRATION MEANS REDUCED COSTS

While the number of users accessing ASE databases may increase, there are fewer IT resources available to administer database servers. For this reason, it is important to simplify and centralize administration, which in turn-minimizes the amount of work to keep users productive and the server running and secure.

CENTRALIZED LOGIN AUTHORITY AND POLICIES

With a centralized login authority, there is one set of policies for a security officer to focus on, one set of password criteria for users to learn and conform to, and one location for upgrades and fixes related to passwords. LDAP Directory Servers are an established way to accomplish this centralization, especially in a heterogeneous environment that may include Windows and multiple Unix variants.

When a new person is added to a company roster without a central directory server, it could take many independent actions by trained IT professionals to add accounts for the person on all the operating systems and applications that the new person needs to be productive. Consider large populations with tens or hundreds of millions of users. It is unfeasible to administer this population without some automation. Of course, the IT department can create a custom tool to add or maintain the accounts, but this would be yet another tool that an already overwhelmed IT department would need to continually modify and support. Either by hand or by custom tools, expertise is required to accomplish a fundamentally simple administration chore. LDAP solves this with one central mechanism, a set of tools, and an industry-standard protocol for accessing the services provided.

SEPARATION OF ROLES FOR PRIVILEGED USERS

More than just making administration easier, LDAP recognizes that separation of roles is an important aspect of any secure computing environment. It is often the case that the skill set and security privileges needed to add a new user to the operating system differ from the skill set and privileges needed to add a new database user. It makes good sense to keep an operating system super-user from having the ASE roles needed to access sensitive database data.

ONE PASSWORD TO REMEMBER

A user with one password to remember for operating system and application access is less likely to do non-secure things like write their password down, choose an easy password to crack, or bother IT with requests to reset a seldom used and therefore forgotten password. Users will appreciate a centralized password authority to deal with the complexities of today's computing environments.

LDAP USER AUTHENTICATION – TECHNICAL DETAILS

OVERVIEW

The LDAP User Authentication feature is available with the Security and Directory Services Package, a separately chargeable package for ASE. This feature requires configuration changes to be performed on the ASE. There are no changes required in the client applications to take advantage of this feature. Existing client applications continue to send the username/password information to ASE. ASE authenticates the username/password with the LDAP server instead of syslogins. Authentication using the LDAP server allows the use of enterprise-wide passwords instead of ASE or application specific passwords.

There are two authentication algorithms available for use with LDAP User Authentication; they differ in how a user's Distinguished Name (DN) is obtained. The algorithms use a:

- *Composed* DN for authentication (available starting in ASE 12.5.1), or a
- *Searched* DN for authentication (available starting in ASE 12.5.2).

The primary data structure used with the LDAP protocol is the LDAP URL. ASE uses LDAP URLs to specify an LDAP Server and search criteria to use to authenticate login requests.

COMPOSED DN ALGORITHM

The following table describes the login sequence when the Composed DN Algorithm is used:

CLIENT	ASE	LDAP SERVER
1) Open Client connects to ASE listener port.		
	2) ASE listener accepts connection.	
3) OpenClient sends internal login record.		
	4) ASE reads login record.	
	5) ASE binds to LDAP Server with DN composed from Primary URL and the login name from the login record. The bind also uses the password from the login record.	
		6) LDAP Server authenticates the user, returning success/failure.
	7) If the Primary URL specifies a search, then ASE sends the search request to the LDAP Server.	
		8) LDAP Server returns the results of the search.
	9) ASE accepts or rejects login, based on results.	

[Table 1] *Login Sequence with Composed DN Algorithm*

Notes: The difference between authentication with LDAP and authentication with the ASE algorithm that uses the *syslogins* system table is the functionality beginning in Step 5.

SEARCHED DN ALGORITHM

The following table describes the login sequence when the Searched DN Algorithm is used:

CLIENT	ASE	LDAP SERVER
1) Open Client connects to ASE listener port.		
	2) ASE listener accepts connection.	
3) OpenClient sends internal login record.		
	4) ASE reads login record.	
	5) ASE binds to LDAP Server with directory server access account.	
		6) LDAP Server authenticates the account, returning success/failure.
	7) ASE sends search request to LDAP Server based on login name from login record and the DN lookup URL.	
		8) LDAP Server returns the results of the search.
	9) ASE reads results to obtain value of attribute from DN lookup URL.	
	10) ASE uses the value of attribute as the DN and the password from the login record to bind to LDAP Server.	
		11) LDAP Server authenticates the user, returning success/failure
	12) If the Primary URL specifies a search, ASE sends the search request to the LDAP Server.	
		13) LDAP Server returns the results of the search.
	14) ASE accepts or rejects login, based on results.	

[TABLE 2] *Login Sequence with Searched DN Algorithm*

Notes: The connection established in steps 5 and 6, above, may persist between ASE authentication attempts; this persistence is an optimization to reuse the connection to do the DN searches.

The administrator may choose to skip steps 12 and 13, above, by not specifying search criteria in the Primary or Secondary URL strings. When criteria are not specified in the Primary or Secondary URL strings, the authentication completes with the success or failure returned by step 11.

Sybase ASE reports a generic login failure to the client if any of these authentication criteria are not met.

WHAT IS AN LDAP URL?

The LDAP URL acronym stands for Lightweight Directory Access Protocol Uniform Resource Locator. An LDAP URL specifies a set of objects or values on an LDAP server. ASE uses LDAP URLs to specify an LDAP Server and search criteria to use to authenticate login requests.

The LDAP URL uses the following syntax:

ldapurl::=ldap://*host:port/node/*?*attributes*?base | one | sub?*filter*
WHERE
HOST is the host name of the LDAP sever.
PORT is the port number of the LDAP server.
NODE specifies the node in the object hierarchy at which to start the search.
ATTRIBUTES is a list of attributes to return in the result set. Each LDAP server may support a different list of attributes.
BASE | ONE | SUB qualifies the search criteria. base specifies a search of the base node; one specifies a search of node and one sublevel below node; and sub specifies a search of node and all node sublevels.
FILTER specifies the attribute or attributes to be authenticated. The filter can be simple, such as "uid=*," or compound, such as "(uid=*)(ou=*group*)."

ENABLING LDAP USER AUTHENTICATION IN ASE

LDAP User Authentication can be enabled by:

1. Setting the configuration parameter "*enable ldap user auth*".
2. Choosing the algorithm to use for authentication and specifying the LDAP URL strings and access account information.
3. Administering user accounts on the LDAP server and ASE.

SETTING CONFIGURATION PARAMETER

A new ASE configuration parameter "enable ldap user auth" is added for this feature. System Security Officer (SSO) role is needed to change the value of this configuration parameter. This parameter is dynamic; it will take effect upon completion of the sp_configure command. Valid values for the configuration parameter are:

VALUE	DESCRIPTION
0	This is the default setting. Indicates LDAP User Authentication is not enabled.
1	ASE searches the LDAP server to verify existence of a user account and will authenticate a user login with the results. In the event that LDAP authentication fails, ASE will fail back to use of syslogins to verify existence of the user account and to authenticate passwords. This level is used to aid with migration of users from ASE authentication to LDAP authentication.
2	ASE searches the LDAP server to verify existence of a user account and will authenticate a user login with the results. Existence of the user in the LDAP Server is required. This setting is required to allow the LDAP server to prevent existing users from authenticating with ASE when they do not have valid LDAP accounts. In security conscious installations, this should be the level used, since it prevents an ASE administrator from creating a new account with sp_addlogin.

[TABLE 3] *Valid values for configuration parameter "enable ldap user auth"*

LDAP Administration

In order to create and maintain LDAP URL search strings and administrative access account information, the stored procedure sp_ldapadmin is used. The SSO role is required to execute the sp_ldapadmin stored procedure. The syntax for this procedure is:

```
sp_ldapadmin  {
        set_primary_url, 'ldapurl'   |
        set_secondary_url, { 'ldapurl' | null } |
        set_access_acct, account_distinguished_name,
               account_password |

        set_dn_lookup_url, ldapurl |
        list_urls |
        list_access_acct |
        check_url, 'ldapurl' |
        check_login, 'login_name' }
```

The ASE sub-commands to sp_ldapadmin are:

SET_PRIMARY_URL specifies the primary LDAP URL search string.
SET_SECONDARY_URL specifies the secondary LDAP URL search string, for failover when primary URL LDAP server is down.
SET_ACCESS_ACCT allows administrative searches using the specified distinguished name and password.
SET_DN_LOOKUP_URL specifies an LDAP URL in which to search for the DN. Setting this URL causes ASE to use *Searched DN Algorithm* for authentication, otherwise, the *Composed DN Algorithm* is used.

LIST_URLS displays LDAP URL search strings.

LIST_ACCESS_ACCT displays LDAP access account DN set via set_access_acct sub-command.
CHECK_URL verifies an LDAP URL search string to be syntactically correct and verifies existence of an LDAP server.
CHECK_LOGIN verifies a user account on the LDAP server using the LDAP URL search string values and access account that may be specified. It does not authenticate the specified user.

COMPOSED DN EXAMPLES

When a simple LDAP server topology and schema are used, the Composed DN Algorithm can be used for user authentication in ASE. If *out-of-the-box* schemas for users are used with iPlanet Directory Servers or OpenLDAP Directory Servers, users are created as objects in the same container in the LDAP server tree and the DN for a user can be derived from the object's location. For this algorithm to work, several restrictions are placed on the LDAP server schema:

- the filter must be specified with the attribute name that uniquely identifies the user to be authenticated,
- the filter is specified with the attribute name=*. The * is the wildcard. The appropriate attribute name to use in the filter depends on the schema used by the LDAP server,
- the ASE login name is the same as the short username, like a UNIX username,
- the DN uses the short username rather than a full name with embedded spaces or punctuation, e.g. 'jqpublic' meets the restriction for a DN, but 'John Q. Public' in the distinguished name does not.

For example, ‘uid=*’ is used as a filter in the iPlanet example below. To compose the DN, ASE replaces the ‘*’ wildcard with the ASE login name to be authenticated, appends the resulting filter to the *node* parameter in the LDAP URL. The resulting DN is ‘uid=myloginname,ou=People, dc=mycompany,dc=com’. After a successful bind operation, ASE uses the connection to search for attribute name *uid* equal to the login name.

```
sp_ldapadmin set_primary_url,
‘ldap://myhost:389/ou=People, dc=mycompany,
dc=com??sub?uid=*’
```

The following example uses the schema defined in OpenLDAP 2.0.25, it uses attribute name “cn”. The composed DN is ‘cn=myloginname,dc=mycompany,dc=com’:

```
sp_ldapadmin set_primary_url,
'ldap://myhost:389/dc=mycompany,dc=com??sub?cn=*'
```

Note that LDAP vendors may use different object names, schema, and attributes than the examples above. Sites may also extend schemas locally or use them in ways different from each other; there are many LDAP URL search strings that are possible and valid.

SEARCHED DN EXAMPLES

To use an Active Directory server or other LDAP server environment that does not meet the restrictions to use the Composed DN Algorithm, the Searched DN Algorithm is used. The following example is for an Active Directory server using out-of-the-box user schema found in Windows 2000 Server:

1. Set the access account information:

```
sp_ldapadmin set_access_acct,
'cn=Admin Account, cn=Users, dc=mycompany, dc=com',
'Admin Account secret password'
```

2. Set the primary URL:

```
sp_ldapadmin set_primary_url,
'ldap://hostname:389/'
```

3. Set the distinguished name lookup URL search string:

```
sp_ldapadmin set_dn_lookup_url,
'ldap://hostname:389/cn=Users,dc=mycompany,dc=com?dist
inguishedName?one?samaccountname=*'
```

In Windows 2000, the short name is often referred to as the “User Logon Name” and is given attribute name 'samaccountname' in the default schema, this is the attribute name used to match the ASE login name. The DN for a user contains a full name with punctuation and embedded spaces, e.g. 'cn=John Q. Public, cn=Users, dc=mycompany, dc=com'. The DN on Windows does not use the short name. Therefore, the Searched DN Algorithm is appropriate for sites using (default) Active Directory schema for their LDAP server. Note also that the primary URL does not specify a search; it relies just on the bind operation for the authentication.

EXAMPLES USING SEARCH FILTERS TO RESTRICT ASE ACCESS

LDAP URL search strings can also be used to restrict access to subsets of the users found in an LDAP server. For example, to restrict logins to users that are only in an accounting group, a compound filter is used to restrict access to the subset of users where attribute group=accounting. The LDAP URL string to do this using the earlier Composed DN Algorithm example for an iPlanet server is:

```
sp_ldapadmin set_primary_url,
'ldap://myhost:389/ou=People,dc=mycompany,
dc=com??sub?(&(uid=*)(group=accounting))'
```

ASE will bind with DN "uid=mylogin,ou=People,dc=mycompany,dc=com". Then, after successfully binding with this identity, it will search for "ou=People,dc=mycompany,dc=com??sub?(&(uid=mylogin)(group=accounting))". If this search returns more than 0 objects, then authentication succeeds.

Below are a few more examples of LDAP URL strings containing compound filters.

```
sp_ldapadmin set_primary_url,
'ldap://myhost:389/ou=people,dc=mycompany,dc=com??sub?
(&(uid=*)(ou=accounting) (l=Santa Clara))'

sp_ldapadmin, set_primary_url,
'ldap://myhost:389/ou=people,dc=mycompany,dc=com??sub?
(&(uid=*)(ou=Human%20Resources))'
```

ASE LOGIN AND LDAP USER ACCOUNT

Once LDAP User Authentication is enabled, an authentication algorithm and URL strings are chosen and set, there remains some decisions to be made for user accounts. The LDAP administrator will create and maintain accounts in the LDAP server, and the DBA will create and maintain accounts in the ASE. Alternatively, the DBA can choose administration options that allow flexibility with login accounts when integrating ASE with external authentication mechanisms such as LDAP server. Enhanced Login Controls, introduced in ASE 12.5.2 and described in a later heading, offers these new login controls. The DBA continues to administer the ASE account roles, default database, default language, and other login-specific attributes using traditional commands and procedures.

The following table describes updates to syslogins table ASE makes at login time. It assumes that DAP User Authentication is configured.

ROW EXISTS IN ASE SYSLOGINS	LDAP SERVER AUTHENTICATION SUCCEEDS	RESULTING CHANGES IN SYSLOGINS
NO	YES	No change, login fails.
NO	NO	No change, login fails.
YES	YES	Update row if password has changed.
YES	NO	No change.

[TABLE 4] *Update Actions to syslogins*

ENHANCED LOGIN CONTROLS

ASE version 12.5.2 introduces new ways to control authentication at the granularity of a login. These options provide added flexibility in managing user accounts.

With the first mechanism SSO can force a login to use ASE authentication even though the rest of ASE logins are set up for LDAP authentication. For example, nightly batch jobs can use syslogins authentication compared to end-user login authenticated with LDAPUA.

```
          sp_modifylogin "nightlyjob", "authenticate with", "ASE"
```

The other mechanism allows SSO to specify action to take in case the LDAP user entry does not exist in syslogins already. The action can be to either create the login, or to map the LDAP user to an already existing login. For example, an external user "jsmith" can be mapped to Adaptive Server user "guest" by using the following command:

```
          sp_maplogin NULL, "jsmith", "guest"
```

Once authenticated, "jsmith" gets the privileges of "guest". The audit login record shows both the external client username and the Adaptive Server user name.

The following example tells ASE to create a new login for all external users authenticated with LDAP, in case a login does not already exist:

```
sp_maplogin LDAP, NULL, "create login"
```

CONFIGURATION TIPS

Configuring LDAP User Authentication in New ASE Installations

New server installations need to:

1. Specify LDAP URL search strings and access account values in ASE.
2. Set the configuration parameter "enable ldap user auth" to 2.
3. Add users in the LDAP directory server using LDAP vendor supplied tools.
4. Add users to ASE using sp_addlogin or by using sp_maplogin to automatically create login accounts upon authentication or apply other login controls.

Migrating Existing ASE Installations to LDAP User Authentication

In order to avoid disruption of service in existing server installations, the following procedure may be followed:

1. Specify an LDAP URL search string to ASE
2. Set the configuration parameter "enable ldap user auth" to 1
3. Add users in the LDAP directory server
4. When all users are added to the LDAP server, set the configuration parameter "enable ldap user auth" to 2 to require all authentications to be done with LDAP User Authentication, or use sp_maplogin to override configuration parameters with more finely settable login controls.

FAILOVER SUPPORT

When a major failure occurs in the LDAP Directory Server specified by the primary URL and the server no longer responds to network requests, ASE attempts to connect to the secondary LDAP Directory Server specified by the secondary URL. ASE uses the LDAP protocol function ldap_init() to determine if a connection to the LDAP Directory Server can be opened. A NULL URL string, or otherwise syntactically invalid primary URL string, will also cause ASE to attempt failover to a secondary URL. ASE does not failover to the secondary URL for failures returned by the LDAP protocol bind or search operations.

Secondary LDAP URL may be specified using the sp_ldapadmin command.

```
sp_ldapadmin, set_secondary_url,
'ldap://backuphost:389/ou=people,dc=mycompany,dc=com??
sub?uid=*'
```

CONCLUSION

LDAP User Authentication is ideal for organizations with an existing computing environment who want to simplify and centralize user administration, or for those in a new computing environment who just want to avoid unnecessary complexities for administering users. LDAP User Authentication works with Directory Servers that meet LDAP Version 3 of the protocol standard, including Active Directory, iPlanet, and OpenLDAP Directory Server.

SYBASE®

Sybase Incorporated
Worldwide Headquarters
One Sybase Drive
Dublin CA, 94568 USA
T 1.800.8.SYBASE
F 1.925.234.5678
www.sybase.com